UNIVERSITÀ DEGLI STUDI DI TORINO

DIPARTIMENTO DI MATEMATICA GIUSEPPE PEANO

SCUOLA DI SCIENZE DELLA NATURA

Corso di Laurea Magistrale in Matematica

Tesi di Laurea Magistrale

Comparative correlation analyses of high-dimensional point processes: applications to neuroscience

Relatore: Cristina Zucca

Candidato: Pietro Quaglio

Correlatori: Sonja Grün, Emiliano Torre

ANNO ACCADEMICO 2014/2015

Contents

Introduction

Modern extracellular recording techniques allow to record simultaneously from as many as 100 or more neurons *in vivo* [9]. This allows one to observe the concerted spiking activity of relatively small (but increasingly larger) neural populations during behavior in various animal species. Thereby, not just marginal behavior from individual cells but collective phenomena as well are getting increasingly accessible to neuroscientists. Making sense out of these complex high-dimensional data requires suitable analysis tools. In particular, ad-hoc correlation analyses have been developed in the past years to reveal different types of correlation structures in massively parallel spike trains, and have been proven through simulations to yield high performance when the addressed correlation structure actually characterizes the data. However, performance drops and results become apparently contraddictory when different correlation structures, or a mixture of correlation models, come into place - as it is likely to be for electrophysiological data The present work addresses this issue by A) presenting a generalized version of existing stochastic models of simultaneous spiking activitywhich allows flexible correlation structures to be specified, B) using this model to generate simulated data with diverse correlation structures, C) analyse these data with two of the above-mentioned correlation analyses (CuBIC [42] and SPADE [43]), and D) comparing and combining the results to better describe the real correlation structure underlying the data as compared to what individual results allow one to do.

This workflow, here demonstrated on the two mentioned analyses, can be extended to different methods(e.g. [18, 5]).

In Chapter 1 we present the general mathematical instruments used to construct the models and their generative procedure. In Chapter 2 we show the relation between point processes and spike train data and briefly discuss the relevance of high(er) order correlations in neuronal encoding [20, 3, 33].

Ini Chapter 3 we present the correlated Multidimensional Poisson Process (MPP) used to model and simulate correlated spike trains and introduce the algorithmsto generate simulated data for each parameter setting of the model.

Chapter 4 presents the two statistical methods considered in the comparative analysis [42, 43].

In Chapter 5 we finally use the MPP model to simulate spike trains in which different correlations structure are embedded, and use these data in order to proceed with the comparative analysis of the methods.

Chapter 1

Mathematical Background

1.1 Point processes

Point processes are stochastic processes widely used in probability and statistics to model series of events [10], [11], where each event occurs randomly in time according to certain statistical rules. The realization of a point process is a collection of such events which follow each other. The set of time points where an event can possibly fall is the domain of the point process. In this thesis point processes are used to model events which are continuous in time, so that the domain of the process is the real space $\mathbb{R}$.

Another useful representation of a point process is the interval representation, which associates to each event its time difference with the preceeding event. Thus, point processes can be equivalently represented in terms of its stochastic events t_i or its intervals $\{\Delta_i\}$, $\Delta_i := t_i - t_{i-1}$, and the statistical laws regulating the time events t_i can be mapped to those of the time differences Δ_i.

The point processes are used to model random events succeeding in time, such as the arrival of customers in a queue, emissions from a radioactive source, occurrence of calamities and in general every sequence of events that can be considered a random discrete sequence of points in time.

In this perspective we will use point processes to model spiking activity from neural cells, as commonly done by the scientific community ([25], [7]). The details of this formalization are explained in Chapter 2.

Since we will intensively use stochastic point processes throughout this thesis, we will introduce in this chapter the theoretical background necessary to later on move to models specific to neuroscience applications.

A particular realization (trajectory) of a point process can be described as a function of time t with many different formal representations.

The counting representation consists of a function of time $N(t)$, that, fixed a domain $[t_{start}, t_{stop}] \in [-\infty, +\infty]$, defined as:

$$N(t) := \text{number of events occured in the interval } [t_{start}, t_{stop}] = |T|$$

where the module $|.|$ is simply the number of elements of the set T, that is the set of time-points $T = (t_1, \ldots, t_n)$ of the particular realization of the process such that $t_{start} \leq t_i \leq t$, so that $N(t) = n$.

Another representation, commonly used in neuroscience, is given by the time-point events. Again it consists of a function of time $X(t)$ that, given the set of times $T = (t_1, \ldots, t_n)$ in which occurred an event in the interval $[t_{start}, t_{stop}]$, is defined as:

$$X(t) = \begin{cases} 1 & \text{if } t = t_i \text{ for } i = 1, \ldots, n \\ 0 & \text{otherwise} \end{cases}$$

$$= \sum_{i=1}^{n} \delta(t - t_i)$$

where δ is the Dirac's delta function:

$$\delta(t) := \begin{cases} 1 & t = 0 \\ 0 & otherwise \end{cases}$$

This two representations are equivalent:

$$X(t) = \lim_{\Delta t \to 0} (N(t + \Delta t) - N(t)) = \begin{cases} 1 & \text{if an event occurred at time } t \\ 0 & \text{otherwise} \end{cases}$$

Equivalently it is possible to derive the count representation from the time-point representation by:

$$N(t) = \int_{t_{start}}^{t_{stop}} X(t) = \int_{t_{start}}^{t_{stop}} \sum_{i=1}^{n} \delta(t - t_i) = n$$

Different representations of a point process can become useful in different circumstances, as will become apparent throughout this work.

We will now introduce a particular point process, i.e. the Poisson process.

1.1.1 Poisson process

There are two definitions of a Poisson process that can be shown to be equivalent (as shown in [10]).

Definition 1.1.1. A stationary Poisson process $N(t)$ of parameter $\lambda \in \mathbb{R}$ is a point process which satisfies:

1. $N(t+\Delta t) - N(t) \sim Poisson(\lambda \Delta t)$ for any interval $(t, t+\Delta t]$, or equivalently $\mathbb{P}(N(t + \Delta t) - N(t) = n) = \frac{(\lambda \Delta t)^n e^{-\lambda \Delta t}}{n!}$
2. For any non-overlapping intervals $(t_1, t_2]$ and $(t_3, t_4]$ the two random variables $\Delta N((t_1, t_2]) = N(t_2) - N(t_1)$ and $\Delta N((t_3, t_4]) = N(t_4) - N(t_3)$ are independent

An equivalent definition of a Poisson process is the following.

Definition 1.1.2. A stationary Poisson process $N(t)$ of parameter $\lambda \in \mathbb{R}$ is a point process which satisfies:

1. $\mathbb{P}(N(t+\Delta t) - N(t) = 1) = \lambda \Delta t + o(\Delta t)$ for any interval $(t, t+\Delta t]$
2. $\mathbb{P}(N(t+\Delta t) - N(t) > 1) = o(\Delta t)$ for any interval $(t, t+\Delta t]$
3. For any non-overlapping intervals $(t_1, t_2]$ and $(t_3, t_4]$ the two random variables $\Delta N((t_1, t_2]) = N(t_2) - N(t_1)$ and $\Delta N((t_3, t_4]) = N(t_4) - N(t_3)$ are independent

We now introduce three properties that make the Poisson process analytically treatable.

Proposition 1.1.1.
Given a Poisson process $N(t)$ the expected number of events at time t is given by:

$$\mathbb{E}(N(t)) = \lambda(t - t_{start})$$

Indeed the parameter λ of the process is commonly called the rate parameter since it represents the mean number of events per time unit.

Proposition 1.1.2.
The inter-event intervals of a Poisson process $N(t)$ are i.i.d exponential random variable of parameter λ (i.e. $\mathbb{P}(\text{inter-event interval} \leq \tau) = \mathbb{P}(N(t+\tau) - N(t) \neq 0) = e^{-\lambda\tau}$)

Proposition 1.1.3.
If $N(t)$ is a Poisson process and assumed $N(t) = n$, all time points $(t_1, \ldots, t_n)$ with $t_i \in (t_{start}, t]$ are uniformly distributed in the interval $(t_{start}, t]$.

For detailed demonstrations of these properties we refer to [10, 11]
The third property suggests a trivial technique to generate a Poisson process within a finite interval $[0, \bar{t}]$:

Algorithm 1.1.1.

1. Generate a Poisson random variable $N(\bar{t})$ of parameter $\lambda\bar{t}$
2. Sample $N(\bar{t})$ realizations of the uniform random variable in the interval $[0, \bar{t}]$
3. The order vector $(u_1, \ldots, u_{N(\bar{t})})$ is the time event trajectory of the Poisson process

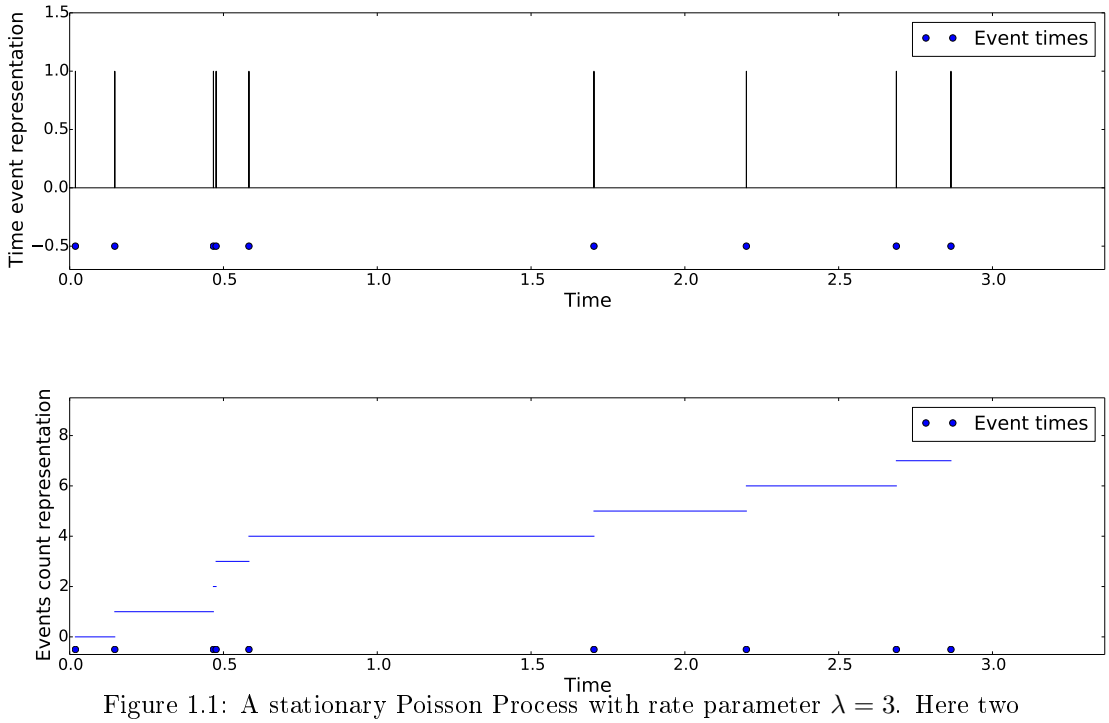

Figure 1.1: A stationary Poisson Process with rate parameter $\lambda = 3$. Here two possible representations are presented: binary representation $X(t)$ (top) and count representation $N(t)$ (bottom)

1.1.2 Non-stationary Poisson process

The Poisson process as introduced above has events whose occurrence probability is stationary over time (equivalently: the inter-event intervals are identically distributed). This stochastic process allows a generalization to occurrence probabilities which are non-stationary over time by making the rate parameter λ a function of time:

Definition 1.1.3. Non-stationary Poisson Process

Given a continuous (or piece-wise continuous) function of time $\lambda(t) : \mathbb{R} \to \mathbb{R}$, a non-stationary Poisson process with rate parameter $\lambda(t)$ is a point process such that:

1. $N(t+\Delta t) - N(t) \sim Poisson(\int_t^{t+\Delta t} \lambda(s)ds)$ for any interval $(t, t+\Delta t]$, or equivalently $\mathbb{P}(N(t+\Delta t) - N(t) = n) = \frac{(\int_t^{t+\Delta t} \lambda(s)ds)^n e^{\int_t^{t+\Delta t} \lambda(s)ds}}{n!}$
2. For any non-overlapping intervals $(t_1, t_2]$ and $(t_3, t_4]$ the two random variables $\Delta N((t_1, t_2]) = N(t_2) - N(t_1)$ and $\Delta N((t_3, t_4]) = N(t_4) - N(t_3)$ are independent

Equivalently to the stationary case we can introduce a proposition regarding the expected value of the process and for the proof we refer to [11]:

Proposition 1.1.4.

The expected value of a non-stationary Poisson process with rate function $\lambda(t)$ at time time is given by:

$$\mathbb{E}(N(t)) = \int_{t_{start}}^{t} \lambda(s)ds$$

This explains the fact that the function $\lambda(t)$ is called rate function or rate profile function.

Unlike the case of a stationary Poisson process, a non-stationary Poisson process does not allow a definition of inter-event distribution. This implies that the technique described before to generate a stationary Poisson process cannot be generalized to the non-stationary case. For this reason we will now introduce a method to generate a Poisson process with bounded rate $\lambda(t)$ in an interval $(t_{start}, t_{stop}]$:

Algorithm 1.1.2.

1. Set $\hat{\lambda} = \max_{(t_{start}, t_{stop}]} \lambda(t)$
2. Generate a stationary Poisson process $N_{stat}(t)$ with constant rate $\hat{\lambda}$ in the interval $(t_{start}, t_{stop}]$
3. For each time point $\bar{t} = (t_1, \ldots, t_{N(t_{stop})})$ generate a uniform variable in the interval $[0, 1]$, obtaining the realization $(u_1, \ldots, u_{N(t_{stop})})$

4. The series of events of the non-stationary process $N(t)$ is the subset of $\bar{t}$ such that $\{t_i | u_i \geq \frac{\lambda(t_i)}{\hat{\lambda}}\}$, or equivalently accept each time point t_i of the stationary process with probability equal to $\frac{\lambda(t_i)}{\hat{\lambda}}$

This technique is commonly called thinning and the process it yields is indeed a non-stationary Poisson process with rate function $\lambda(t)$.

We introduce now a general result for the Poisson random variable that we will use to show that the algorithm 1.1.2 generates the desired process.

Proposition 1.1.5. Suppose X is a random variable with a Poisson distribution with mean λ. Suppose that for all $n \geq 1$, conditional on $X = n$, each of the n is independently labeled as being of type 1 or 2 with probability $p, 1-p$ respectively, and let X_i denote the number of type i,$i = 1, 2$ (in particular $X = X_1 + X_2$). Then the two X_i are independent random variables, and X_1 has a Poisson distribution with mean λp and X_2 has a Poisson distribution with mean $\lambda(1-p)$

We can now prove that the algorithm produces a non-stationary Poisson process.

Proposition 1.1.6.

A process $N(t)$ generated with the algorithm 1.1.2 is as non-stationary Poisson process with rate function λ(t).

Proof:

We have to prove the two conditions of the definition of non-stationary Poisson process.

Because of the independence of the increments on non-overlapping intervals of the stationary process $N_{stat}(t)$ and the independence of the thinning of the single event we end up with independent increments also in the non-stationary process

Then we have to show that the random variable $N(t)$ is Poissonian distributed with rate parameter equal to $\int_{t_{start}}^{t} \lambda(s)ds$. We can apply the general result obtained in (1.1.2).

In order to show that the parameter of the partition $N(t)$ of $N_{stat}(t)$ is the desired one we consider:

$$\mathbb{E}(N(t)) = \mathbb{E}(\mathbb{E}(N(t)|N_{stat}(t)) = \mathbb{E}(\sum_{i=1}^{N_{stat}(t)} B_i) = \mathbb{E}(N_{stat}(t))\mathbb{E}(B_i) \qquad (1.1)$$

where the random variables B_i are independent binomial random variables given by

$$B_i = \begin{cases} 1 & p_i = \frac{\lambda(t_i)}{\hat{\lambda}} \\ 0 & 1 - p_i \end{cases}$$

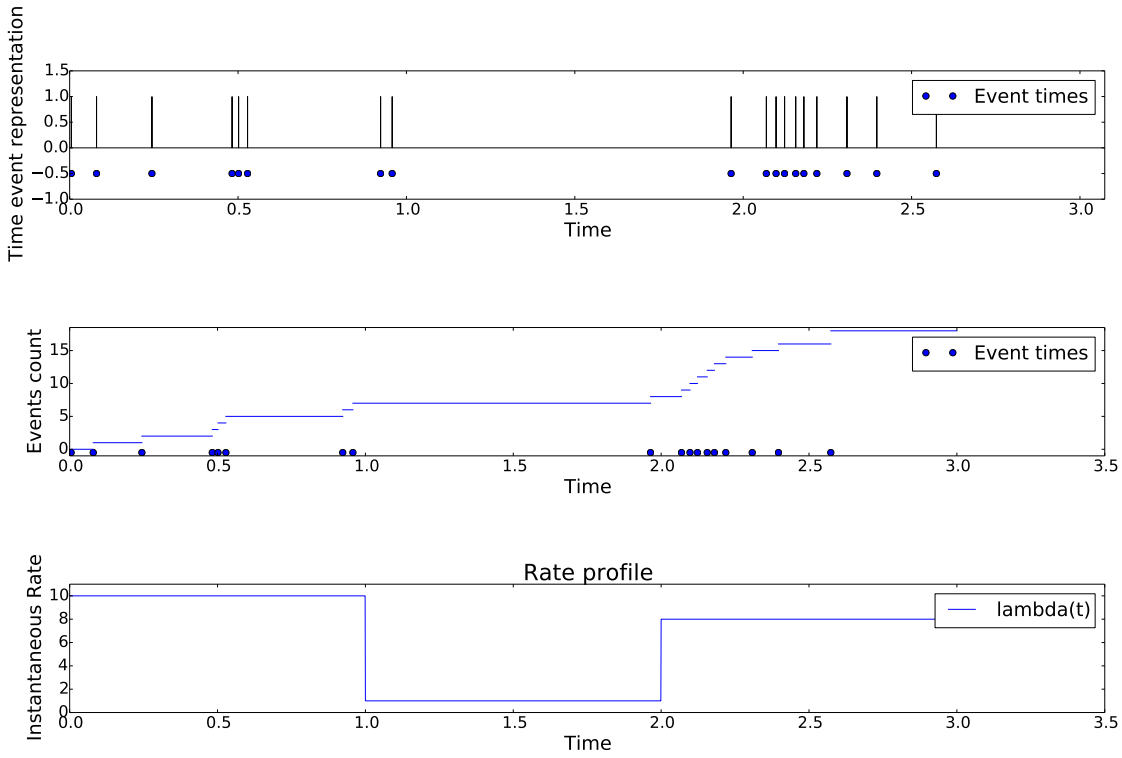

Figure 1.2: A non-stationary Poisson Process with rate parameter $\lambda(t) = 10$ for $t \in (0, 1]$, $\lambda(t) = 1$ for $t \in (1, 2]$, $\lambda(t) = 8$ for $t \in (2, 3]$. Here are presented the two possible representations: binary representation $X(t)$ (top) and count representation $N(t)$ (bottom). In the low section of the plots are represent the times in which the events are occurred

Since we are conditioning on having a given number of total events, all the times t_i are uniformly distributed in the interval $(t, t_{stop}]$ because of the property (1.1.1) of the stationary Poisson process $N_{stat}(t)$. So we can derive from (1.1):

$$\mathbb{E}(N(t)) = \mathbb{E}(N_{stat}(t))\mathbb{E}(\mathbb{E}(B_i|N(t))) = \hat{\lambda}\mathbb{E}(\frac{\lambda(U)}{\hat{\lambda}}) = \hat{\lambda}\int_{t_{start}}^{t} \frac{\lambda(s)}{\hat{\lambda}} ds = \int_{t_{start}}^{t} \lambda(s) ds$$

In the next section we will introduce a general feature of the Point Processes: the conditional intensity function. This mathematical object will be the base to construct a general algorithm for the generation of a generic Point Process.

1.1.3 Generative technique for general Point Processes

In this section we will introduce a more general and computationally efficient technique to generate any class of point process following the approach from [7].

Definition 1.1.4. Conditional intensity function

We consider a point process $N(t)$ defined on an interval $(t_{start}, t_{stop}]$. We define its history at time t as the time-point of the process up to time t:

$$H_t = (t_1, \ldots, t_{N(t)})$$

Then the conditional intensity function of the process $N(t)$ is defined as:

$$\lambda(t|H_t) = \lim_{\Delta t \to 0} \frac{\mathbb{P}(N(t+\Delta t) - N(t) = 1|H_t)}{\Delta t}$$

We can relate the conditional intensity function with the time interval probability density, conditioned on the history of the process.

Proposition 1.1.7. Given a point process with the conditional intensity function $\lambda(t|H_t)$ and a conditioned density of inter-event time intervals $f(t|H_t)$ we have:

$$\lambda(t|H_t) = \frac{f(t|H_t)}{1 - \int_{t_{N(t)}}^{t} f(s|H_t)ds} \tag{1.2}$$

Proof:

$$\mathbb{P}(N(t+\Delta t) - N(t) = 1|H_t) = \mathbb{P}((t_{N(t)+1} \in [t, t+\Delta t)|t_{N(t)+1} > t, H_t) =$$

$$\frac{\mathbb{P}(t_{N(t)+1} \in [t, t+\Delta t)|H_t)}{\mathbb{P}(t_{N(t)} > t|Ht)} = \frac{\int_t^{t+\Delta t} f(u|H_t)du}{1 - \int_{t_{N(t)}}^{t} f(u|H_t)du} \approx \frac{f(t|H_t)\Delta t}{1 - \int_{t_{N(t)}}^{t} f(u|H_t)du}$$

Dividing all terms byΔt and taking the limit to infinity we obtain:

$$\lambda(t|H_t) = \lim_{\Delta t \to 0} \frac{\mathbb{P}(N(t+\Delta t) - N(t) = 1|H_t)}{\Delta t} = \frac{f(t|H_t)}{1 - \int_{t_{N(t)}}^{t} f(u|H_t)du}$$

The conditional intensity function is well defined for every kind of point process [11].

We can now derive analytically the intensity function for the particular case of the Poisson process.

Proposition 1.1.8.
Given a Poisson process $N(t)$ with rate parameter λ the conditional intensity function is:

$$\lambda(t|H_t) = \lambda$$

Proof:
The inter-event times of a Poisson process are exponentially distributed and independent of the process history H_t, hence we have:

$$\lambda(t|H_t) = \frac{f(t|H_t)}{1 - \int_0^t f(u|H_t)du} = \frac{\lambda e^{-\lambda t}}{1 - (1 - e^{-\lambda t})} = \frac{\lambda e^{-\lambda t}}{e^{-\lambda t}} = \lambda$$

Similarly, the intensity and conditional intensity functions are identical for the non-stationary Poisson process with rate function $\lambda(t)$.

Proposition 1.1.9.
Given a non-stationary Poisson process $N(t)$ with rate parameter $\lambda(t)$ its conditional intensity function is:

$$\lambda(t|H_t) = \lambda(t)$$

Proof:
The inter-event times of a Poisson process are exponentially distributed with parameter $\lambda(t)$ and independent of the process history H_t, hence we have:

$$\lambda(t|H_t) = \frac{f(t|H_t)}{1 - \int_0^t f(u|H_t)du} = \frac{\lambda(t) e^{-\lambda(t)}}{1 - (1 - e^{-\lambda(t)})} = \frac{\lambda(t) e^{-\lambda(t)}}{e^{-\lambda(t)}} = \lambda(t)$$

We now prove a lemma that clarifies further the relation between intensity function and the distribution of inter-times. We will need this result in order to prove the theorem on which to construct the generative algorithm for a general point process.

Lemma 1.1.1. *Given a point process with intensity function $\lambda(t|H_t)$ the joint density of inter-event intervals is:*

$$f(t_1, \ldots, t_n \cap N(t) = n) = \prod_{i=1}^{n} \lambda(t_i|H_{t_i}) \exp\left(\int_{t_{i-1}}^{t_i} \lambda(u|H_u)du\right) \exp\left(\int_{t_n}^{t} \lambda(u|H_u)du\right)$$

Where $t_0 = t_{start}$, the starting point of the interval in which is defined the process and equivalently the intensity function.

Proof:
Given a probability density function $f(t|H_t)$ we can associate the correspondent probability distribution:

$$F(t|H_t) := \mathbb{P}(t_{N(t)+1} < t|H_t) = \int_{t_{N(t)}}^{t} f(u)du$$

Or equivalently:

$$\frac{dF(t|H_t)}{dt} = F'(t|H_t) = f(t|H_t)$$

Using this relation in Eq. (1.2) we get a differential equation that can be solved as a function of $\lambda(t|H_t)$:

$$\frac{F'(t|H_t)}{1 - F(t|H_t)} = \lambda(t|H_t)$$

The solution of this differential problem provides us an equation for the distribution $F(t|H_t)$:

$$1 - F(t|H_t) = \exp(\int_{t_{N(t)}}^{t} \lambda(u|H_u)du)$$

We can also state that the definition $\lambda(t|H_t) = \frac{f(t|H_t)}{1-\int_{N(t)}^{t} f(u|H_t)du}$ implies:

$$f(t|H_t) = \lambda(t|H_t) \exp(\int_{t_{N(t)}}^{t} \lambda(u|H_u)du) \tag{1.3}$$

Then the thesis derives from:

$$f(t_1, \ldots, t_n \cap N(t) = n) = f(t_1, \ldots, t_n \cap t_{n+1} > t) =$$

$$= f(t_1, \ldots, t_n \cap N(t_n) = n)\mathbb{P}(t_{n+1} < t|H_t) =$$

$$= \prod_{i=1}^{n} f(t_i)(1 - F(t)) = \prod_{i=1}^{n} \lambda(t_i|H_{t_i}) \exp(\int_{t_{i-1}}^{t_i} \lambda(u|H_u)du) \exp(\int_{t_n}^{t} \lambda(u|H_u)du)$$

With these results we can now move to the main theorem of this chapter, that allows to relate any general point process to the standard Poisson process:

Theorem 1.1.2. ***The time rescaling theorem***

Let $t_{start} < t_1 <, ..., < t_n < t_{stop}$ be a realization of a point process with a conditional intensity function $\lambda(t|H_t)$ satisfying $0 < \lambda(t|H_t)$ for all $t \in (t_{start}, t_{stop}]$. We define the transformation:

$$\Lambda(t_i) = \int_{t_{start}}^{t_i} \lambda(u|H_u)du$$

for $i = 1, \ldots, n$ and $\mathbb{P}(\Lambda(t) < \infty) = 1$ for all $t \in (t_{start}, t_{stop}]$.

Then the new process defined by the transformation $\Lambda(t)$ is a Poisson process in the interval $(\Lambda(t_{start}), \Lambda(t_{stop})]$ of rate parameter 1.

Proof:

We consider the random variables $\tau_i = \Lambda(t_i) - \Lambda(t_{i-1})$ for $i = 2, \ldots, n$ and $\tau_{t_{stop}} = \int_{t_n}^{t_{stop}} \lambda(u|H_u)du$. Proving the thesis is equivalent to prove that all variables τ_i are independent and exponentially distributed with rate parameter $\lambda = 1$.

Since all the transformations to get the τ_i's are bijective and since $\tau_{n+1} > \tau_{t_{stop}}$ if and only if $t_{n+1} > t_{stop}$ we can write the joint probability density of the τ_is as:

$$f(\tau_1, \ldots, \tau_n \cap \tau_{n+1} > \tau_{t_{stop}}) = f(\tau_1, \ldots, \tau_n)\mathbb{P}(\tau_{n+1} > \tau_{t_{stop}}|\tau_1, \ldots, \tau_n) \quad (1.4)$$

We can now derive the two terms of the right-hand side of this equation. Taking into account that, by definition, these two events are equivalent:

$$\{\tau_{n+1} > \tau_{t_{stop}}|\tau_1, \ldots, \tau_n\} = \{t_{n+1} > t_{stop}|t_1, \ldots, t_n\}$$

This implies that:

$$\mathbb{P}(\tau_{n+1} > \tau_{t_{stop}}|\tau_1, \ldots, \tau_n) = \mathbb{P}(t_{n+1} > t_{stop}|t_1, \ldots, t_n) =$$
$$= \exp(\int_{t_n}^{t_{stop}} \lambda(u|H_u)du) = \exp(\tau_{t_{stop}})$$

For the second term we can use the change-variable-formula ([35]) and consider the transformation that relates the τ_k's and t_k's and its Jacobian matrix J:

$$f(\tau_1, \ldots, \tau_n) = |J|f(t_1, \ldots, t_n \cap N(t) = n) \quad (1.5)$$

where $||$ is the determinant of J. Since for $i = 1, \ldots, n$ τ_i is a function of $t_1, \ldots, t_i$, J is a lower triangular matrix which implies that $|J| = \prod_{i=1}^{n} J_{ii}$. By the assumption $0 < \lambda(t|H_t)$ and the definition of the transformation $\Lambda(t_i)$ it follows that the transformation that maps t in τ is one by one. Therefore, by the inverse differentiation theorem ([36]), the diagonal elements of the Jacobian matrix J are:

$$J_{ii} = \lambda(t_i|H_{t_i})^{-1}$$

Substituting $|J|$ and the result of (1.3) into (1.5) yields:

$$f(\tau_1, \ldots, \tau_n) = \prod_{i=1}^{n} \lambda(t_i|H_{t_i})^{-1} \prod_{i=1}^{n} \lambda(t_i|H_{t_i}) \exp(\int_{t_{i-1}}^{t_i} \lambda(u|H_u)du) =$$

$$\prod_{i=1}^{n} \exp(\Lambda(t_i) - \Lambda(t_{i-1})) = \prod_{i=1}^{n} \exp(\tau_i)$$

Now we have obtained the two terms to be substituted in (1.4):

$$f(\tau_1, \ldots, \tau_n \cap \tau_{n+1} > \tau_{t_{stop}}) = f(\tau_1, \ldots, \tau_n)\mathbb{P}(\tau_{n+1} > \tau_{t_{stop}} | \tau_1, \ldots, \tau_n) =$$

$$= \prod_{i=1}^{n} \exp(\tau_i) \exp(\tau_{t_{stop}})$$

This finally proves that the τ_is are independent and exponentially distributed with mean equal to 1.

The time rescaling theorem has many applications in of point process theory (e.g. for the validation of the model parameters ([7]).

This result proves a powerful tool to generate point processes with any intensity function $\lambda(t|Ht)$ in the interval $(t_{start}, t_{stop}]$, provided that $\tau = \int^t \lambda(u|H_t)du$ can be solved as a function of t through the following algorithm:

Algorithm 1.1.3.

1. Set $t_0 = 0$ and $k = 0$
2. Generate τ_k exponential inter-time from a stationary Poisson process with unitary rate in $(t_{start}, t_{stop}]$
3. Compute t_k as the solution of $\tau_k = \int_{t_{k-1}}^{t_k} \lambda(u|t_0, \ldots, t_{k-1})du$
4. If $t_k > t_{stop}$ stop
5. $k = k + 1$
6. Go to step 2

As already shown, in the case of the Poisson process the intensity function corresponds to the rate function $\lambda(t)$. Hence, considering a Poisson process, the algorithm effectively rescales the time such that the non-stationary process is transformed into a stationary Poisson process with unitary rate. The transformed time is often known as operational-time (e.g. [28]).

1.2 Multi-variate point processes

Up to now the point processes were assumed to be univariate. This is equivalent to assume that all events are of the same type and that it is possible to add them across time. A natural extension to this assumption is to consider the possibility to observe events in time of different types but on the same time axis. There are different options to model such heterogeneity that we will discuss in the following sections.

1.2.1 Multi-dimensional Point Processes

A first possible representation of different sources of stochastic events is what we will term a multi-dimensional point process in which each margin, or dimension, represents events of a particular class.

Its formalization is a n dimensional random vector $X(t) = (X_1(t), \ldots, X_n(t))$. The size n of the vector corresponds to the number of different sources of events that we want to model. Each component is a realization of a different point process on the same time axis. This representation is typically used to represent series of events coming from different sources but of the same class, such as the arrival times of new customers in different parallel queues or, as we will be interested in our discussion, electrical impulses generated by different neurons. The fundamental concept behind this formalization is to have parallel information in time about the same class of events produced by different sources,

As for the case of univariate process it is possible to consider the count representation in which, alternatively to a vector of delta functions in time $X(t)$, we use a vector of jump functions that represents the count of events at each component $N(t) = (N_1(t), \ldots N_n(t))$. All the arguments about the relation between these two representations are still valid:

$$N_i(t) = \int_{t_{start}}^{t_{stop}} X_i(t)$$

At the same time it is also possible to consider the stochastic evolution of the multi-dimensional vector, considering the joint distribution of all events at a given time point. A multi-dimensional point process is said to be marginally independent if all margins are mutually independent:

$$\mathbb{P}(N_1 = k_1, \ldots, N_n(t) = k_n) = \mathbb{P}(N_1 = k_1) \ldots \mathbb{P}(N_n(t) = k_n)$$

Deviations from independence are described by the correlation structure between the margins.

In the next section we will introduce a particular type of correlation (synchronization) for which we will later on develop our models.

1.2.1.1 Synchrony as correlation

As already mentioned, types of correlation between margins model different relations and causal effects between different phenomena. We will focus on a particular kind of dependence: the occurrence of synchronous events across different margins of the multi-dimensional process. This will allow us to model parallel spike activity, as will be discussed in Chapter 2 ([19]). We say that there is zero-lag correlation (or synchrony correlation) if:

$$\mathbb{P}(X_i(t) = 1 | X_j(t) = 1) \neq \mathbb{P}(X_i(t))$$

The correlation at a time t can be either negative, if:

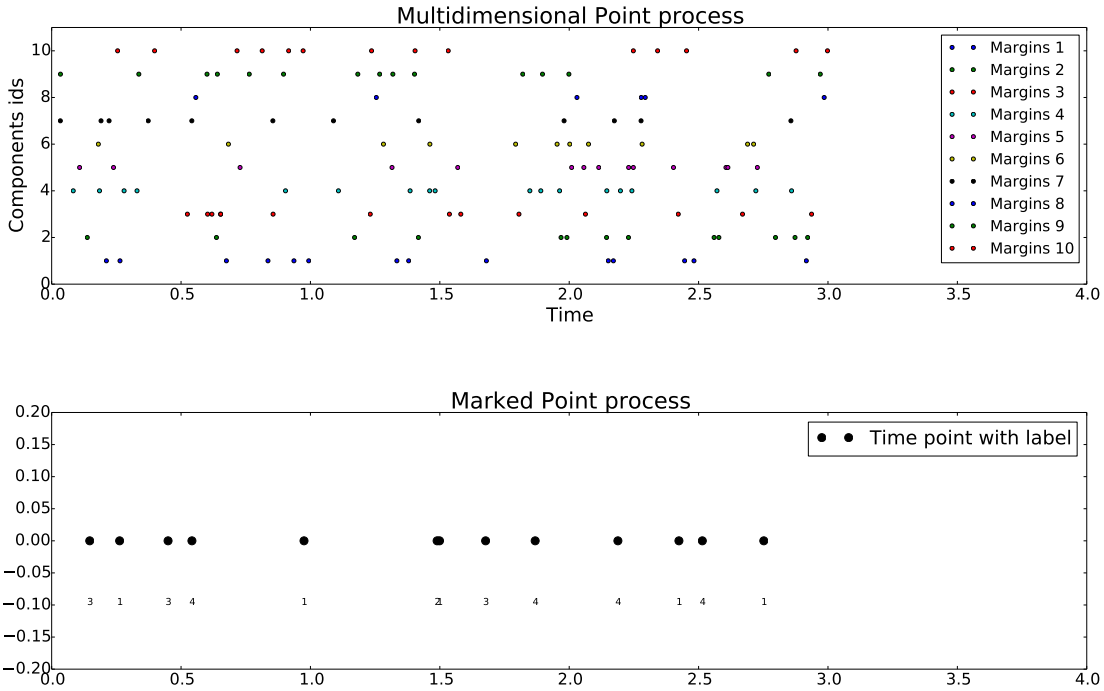

Figure 1.3: Multivariate processes: in the top panel a multi-dimensional process of dimension $n = 10$ with all margins independently Poissonian; in the bottom panel a Marked point process consisting in a Poisson process $X(t)$ with a label D that assume value in the set of integer $(1, 2, 3, 4, 5)$each with uniform probability $\mathbb{P}(D = i) = \frac{1}{5}$

$$\mathbb{P}(X_i(t) = 1 | X_j(t) = 1) \leq \mathbb{P}(X_i(t))$$

or positive, if:

$$\mathbb{P}(X_i(t) = 1 | X_j(t) = 1) \geq \mathbb{P}(X_i(t))$$

We will focus on the second case, where the occurrence of synchronous events is more likely than under independence, where only chance synchronisation occurs. In this context we can now introduce the notion of order of correlation:

Definition 1.2.1. A multi-dimensional point process $X(t) = (X_1(t), \ldots, X_n(t))$ of dimension n has correlation up to order k, with $1 < k \leq n$, if at most k indices $j_1, \ldots, j_k$ exist, such that:

$$\mathbb{P}(X_{j_i}(t) = 1 | X_{j_1}(t) = 1, \ldots, X_{j_{i-1}}(t) = 1, X_{j_{i+1}}(t) = 1 \ldots, X_{j_k}(t) = 1) \neq \mathbb{P}(X_{j_i}(t) = 1)$$

This definition can be interpreted as the maximum number k of margins that we need to consider to assume that the presence of an event at time t is independent from the remaining $n - k$ margins. We can naturally extend this definition by assuming that a multi-dimensional process with all independent margins has order of correlation equal to 1.

1.2.2 Marked point processes

An alternative representation for a set of different event series is the marked point process. This consists of couple of stochastic variables $X_m(t) = (X(t), D)$. To each time events t_i of the process $X(t)$ a realization d_i of the variable D is associated. This can be formally expressed by $X_m(t) = \sum_{i=1}^{N(t)} d_i \delta(t - t_i)$, where $N(t)$ is the count process associated to $X(t)$. This class of processes is thought to model situations in which the source of events is a single point process, but each single event can assume one of a set of (numeric or semantic) values randomly generated (e.g. the arrive of a certain bus at the bus stop, in which the time series represent the delay between a bus and the other and the variable D represent the line to which the bus belongs to).

In this case, as it can be clear already from the example, the correlation would be in the temporal dimension. The distribution of D can present a temporal locking such that a particular realization is more likely in respect to the last value assumed in time (serial correlation). In this discussion we do not focus on this variation, but it is of big interest also in the same context of neurophysiological models.

The relation between the two representations, multi-dimensional and marked processes, is strong and it will be used later on in this work in generative procedure for multi-dimensional correlated process (Section 3.1.1).

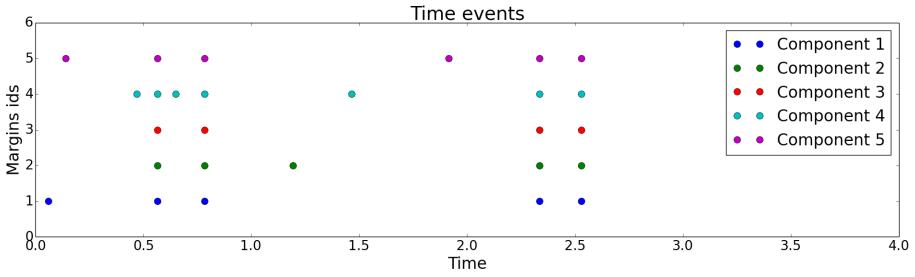

Figure 1.4: An example of multi-dimensional process of size $n = 5$ with all the margins correlated, in this realization we can observe four synchronous occurrence involving all the margins (order of correlation equal to 5)

Chapter 2

Neuroscientific background

In this chapter we will illustrate how spike trains data can be described in terms of point processes, and discuss the open questions in neuroscientific community regarding the correlation structures in parallel recording of neuronal activity. For details we refer to the computational neuroscience literature ([24],[16],[19],[25]). In particular we will illustrate how neurons produce electrophysiological time-point events (action potentials)(Section 2.1), how to model electrophysiological signals with point processes (spike trains) (Section 2.2), how simultaneous recordings are obtained in vivo - leading to a need for multi-dimensional models (section 2.3) and finally why high order correlations in this kind of data are of interest (Section 2.4)

2.1 The firing neuron model (action potential)

In this section we introduce a type of signals that can be recorded from the electric activity of the brain: a sequence of action potentials.

The brain and the spinal cord contain specialized cells called neurons. These cells are the fundamental discrete units that with their dense connection form the network of the central nervous system.

From location to location the neurons differ in their properties and function. However is possible to envisage a paradigm or typical nerve cell with four typical components:

- **Cell body or soma:** this is the focal part from which branching structures emanate. It roughly delineates the input or information-gathering parts of the cell from the output or information-transmitting parts
- **Dendrites:** the dendrites branch several times to form the dendritic trees. They are the chief information-gathering component and many contacts from other cells occur over the dendrites at specialized sites called **synapses**

- **Axon:** the axons are the component along which output electrical signals propagate. The output signal is called **action potential** and it travels through the axon and synapsis to eventually reach target cells
- **Telodendria:** when the axon nears its target cells it branches to form the telodendria. At the end of these, contacts are made with other cells again by means of synapsis

The difference between the extracellular and intracellular potentials is called membrane potential. In absence of activity this potential is negative and fluctuates around -55/-70 mV (although large differences can be observed across cell types and species).

When a synapse or a group of synapses is activated, we see a response of the cell on which the synapses are located that is called synaptic potential. When a synaptic event occurs that decrease the potential difference across the membrane, the neuron is said to be excited or depolirized. At the opposite is possible that when other inputs are activated the potential difference acroos the membrane increase (hyperpolarization), in this case the input and the neuron from which the come are said inhibitory.

These two responses of the neuron that we have considered so far are essentially local responses. The depolarization (or hyperpolarization) is greater in the point where the activated synapse is located and it diminishes as we move away from that point. In contrast to this local response, there is a depolarization of the cell that propagates through the entire length of the axon. This propagating way of depolarization is called **action potential.** What is generally found is that an action potential is elicited when the depolarization of the cell reaches some critical level called **threshold** for action potential generation.

The whole process typically lasts for 1 to 2 ms, during which the membrane potential witnesses an excursion of up to $\sim$100 mV. The action potentials are also known as **spikes** and the cell is said to "fire".

Intracellular recordings from microelectrodes allow to record membrane potentials from individual cells at high frequency (up to tens of KHz), both in vitro and in vivo, and to detect cell spikes as fast and strong negative excursions from the potential's baseline level. A sequence of spikes from the same neurons form a process commonly called **spike train**. Extracellular recordings instead track changes of the electrical potential in the extracellular medium. These changes reflect electrical activity from possibly many cells in the neighborhood of the electrode's position. Fast positive excursions of the signal reflect spiking activity from any of the contributing cells. Under the assumption the spikes fired by the same cell have the same shape when recorded at the same location, spike shapes can be analyzed to assign each action potential shape to one individual cell - a technique known as spike sorting [17, 38, 39].

Spikes are considered to be the elementary bricks of information transmission and processing in the nervous system. Action potentials fired by one neuron travels along its axon to post-synaptic neurons connected to it. The activity of one single neuron is propagated to the surrounding network at different time

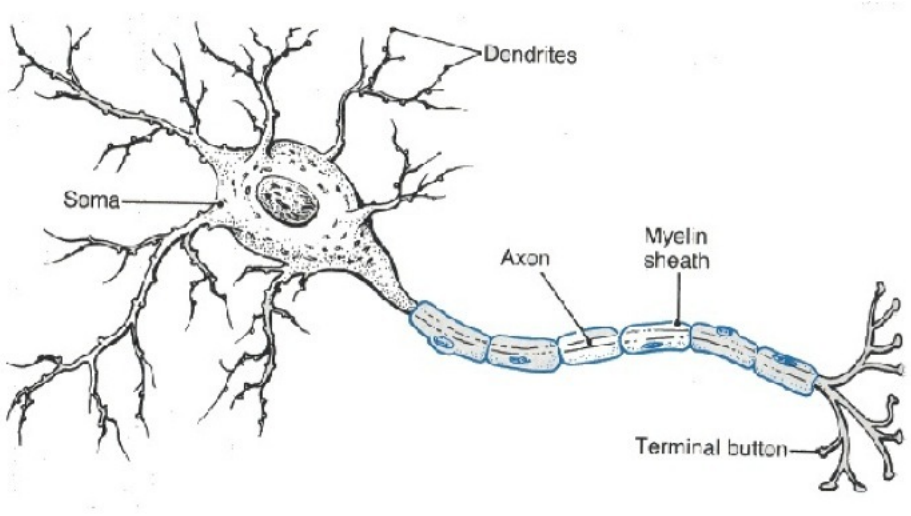

Figure 2.1: A neuron and its principal components (Original source of the figure http://www.neuropsychologysketches.com/Neurons.html)

scales, ranging from several tens of milliseconds to millisecond precision where spatio-temporal spike patterns and precise spike correlations can be observed.

In the next section we present a specific formalism that uses point processes to model a spike train.

2.1.1 Spike train as model for sequence of action potentials

As mentioned above, electrophysiological techniques allow to observe neuronal electrical activity and reconstruct spike trains of individual neurons. Because the shape and the amplitude of action potentials do not carry information, but only their time point occurrence, a point process representation is adequate ([45, 25]). A spike train composed of s spikes recorded in the time interval $[t_{start}, t_{stop})$ can thus be represented as a time series$(t_1, t_2 \ldots, t_s)$ within $(t_{start}, t_{stop}]$. Equivalently, a spike train can be represented by the sequence $\{\Delta t_i\}_{i=2}^{s}$ of all inter-spike intervals (ISIs) $\Delta t_i := t_i - t_{i-1}$ between two consecutive events t_{i-1} and t_i. A meaningful measure that can be extracted from the data is the average firing rate r defined as:

$$r = \frac{\text{number of spikes}}{\text{time inteval of observation}} = \frac{s}{(t_{stop} - t_{start})}$$

Stochastic point processes are a natural model for these time sequences. A spike train $(t_1, \ldots, t_s)$ can be thought of as a realization of a point process $X(t) = \sum_{i=1}^{s} \delta(t - t_i)$, where δ is the Dirac's delta function. Thus:

$$r = \frac{\int_{t_{start}}^{t_{stop}} X(s)ds}{(t_{stop} - t_{start})}$$

Under the assumption of poissonianity the average firing rate corresponds to the definition of rate parameter of a Poisson process. It has to be underlined that this is true only for this specific process, since for a general point process the rate parameter is not even formally defined.

In our further argumentation we will often assume the poissonian hypothesis to construct the stochastic models and the statistical methods. This assumption is in most of the cases necessary to derive the analytics of the mathematical object. At the same time this assumption can hold in few cases for the real neuronal data. The approach has to be data dependent, in the sense that it is possible to test for the Poissonianity of the time series.

For simplicity, in particular of the analytical derivation, we assume here to model and simulate the spike train with Poisson processes.

At the same time it is interesting and probably necessary for the analysis of real datasets to extend the models and the method to non-Poissonian processes and to test the methods' performances neglecting this assumption. However this goes beyond the scope of this work and will be subject of future investigations.

2.2 Parallel recordings of spike trains

Electrophysiological recordings allow to acquire signals simultaneously from different neurons at the same time. The technique advanced in the last years made possible to record from hundred to thousand of neurons simultaneously. The recording devices are composed by different electrodes organized with a grid or array ([31]).

Models for this kind of data are multi-dimensional point processes $X(t) = (X_1(t), \ldots, X_n(t))$, where n is now the number of neurons recorded. The general framework of this class of processes have been introduced in section 1.3.1.

Such a dataset is suitable for many different statistical analyses that are not possible in the case of a single spike train. In particular is possible to develop analysis methods for higher order correlations (HOC)([27, 18, 3, 33]). As already introduced in the previous chapter the possible structures of correlations in a multivariate process can be either either in time or space. Correlation in data can arise by many different mechanisms, here we consider the case of spike synchrony that in terms of point processes, fixed a discretization of time (as we will argue in section 4.1), corresponds to the definition given in section 1.3.1.

In the next section we briefly report why it is of interest to investigate spike synchrony and outline which biological models may actually lead to such correlation structure.

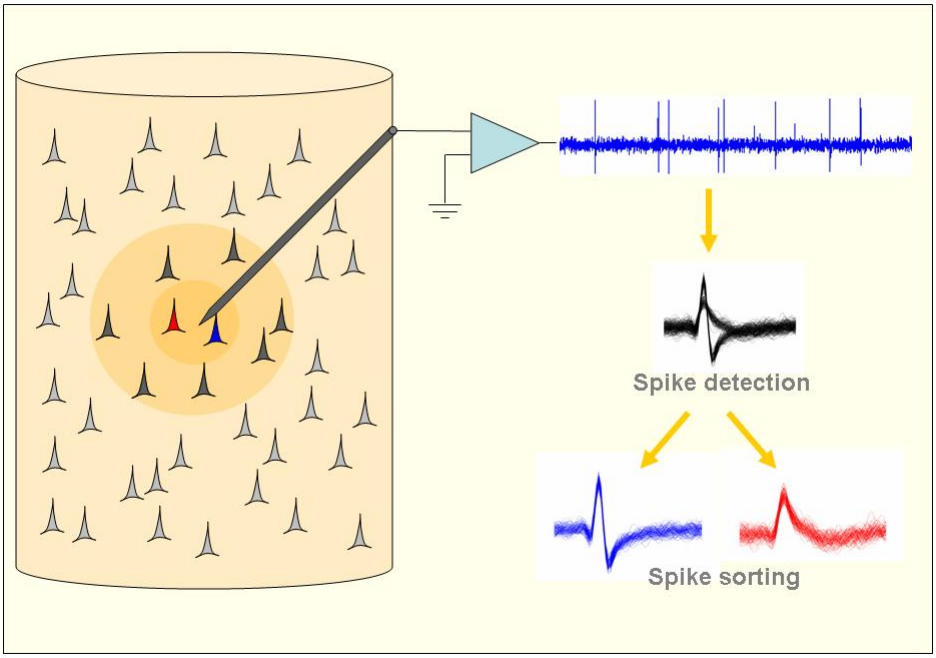

Figure 2.2: Spike recording detection and sorting (original source of the figure: http://www.scholarpedia.org/article/Spike_sorting)

2.2.1 Neural assemblies and synchrony

Already in 1949 Donald Hebb in [20] suggested that neurons coordinate their activity by organizing in functional groups, termed **cell assemblies**. However, whether or not the dynamic formation of cell assemblies constitutes a fundamental principle of cortical information processing remains a controversial issue in current research (e.g. [40]).

It is known that synchronous spike input to receiving neurons are more effective in generating output spikes ([1, 26]) , which leads to the hypothesis that temporal coordination of spiking activity or correlational processing is the defining expression of an active cell assembly. Consequently, approaches to detect assembly activity have focused on detection of correlated spiking. Statistically, coordinating spiking of large neuronal groups has been associated to higher order correlations among the spike trains, where "genuine higher-order correlation" are assumed if coincidence spikes of a neuronal group cannot be explained by the firing rate. This statement in terms of point process coincides with the definition 1.2.1 given in Chapter 1.

In recent years many approaches for the investigation of coordinate activity in parallel spike train were developed, each of which focuses on different aspects:

- methods to to determine the presence of higher order spike correlation without explicitly identifying the participating neurons (e.g. [42, 41, 37])
- methods that directly identify the members of cell assemblies on the basis of the patterns of synchronous spiking activity ([34, 43])
- methods that test whether individual neurons participate in synchronous spiking activity without identifying the groups of correlated neurons ([4])
- methods ([3, 15]) to test for the presence of specific spatio-temporal model (e.g. the synfire chain model that consists in the propagation of spatio-temporal patterns of synchronous spiking activity ([2])

It is clear that each of these method in constructed to detect the presence of assemblies using different statistical measures and assuming different reasons. For this reason we propose here a comparative analysis, focusing on the first two classes of these method. In particular we center the analysis on the methods developed in [41] and in [43].

In order to perform a comparative and integrated analysis of the results of the methods we develop in Chapter 3 a generalized model for multi-dimensional correlated point process that we will use later on to simulate data on which apply the methods.

Chapter 3

Multi-dimensional models of correlated spike trains

In this chapter we introduce the multi-dimensional Poisson process (MPP) as a model of multi-dimensional point process with correlated, Poisson margins. In the context of spike train models, each component of the MPP represents a spike train associated to one neuron, and each event therein represents a spike time of that neuron (See Chapter 2). We start with the case of a MPP where the individual firing rates of all margins are stationary over time (stationary MPP) and identically distributed across margins (homogeneous MPP; see Section 3.1). Then, we generalize to the case of time-stationary, not identically distributed rates (stationary heterogeneous MPP; Section 3.3), and then to a identically distributed, to time non-stationary rates (non-stationary homogeneous MPP; Section 3.4), and finally time non-stationary, not identically distributed rates (non-stationary, heterogeneous MPP; Section 3.5). For each case we derive the relation between the marginal statistics and the correlation structure, and the constraints under which the processes are well defined. We also provide algorithms for the construction of the different types of the MPPs, and relate the model to other models used to model synchronous neuronal activity(Section 3.2).

3.1 Homogeneous stationary multi-dimensional Poisson process

A homogeneous, stationary MPP is a vector of simultaneous Poisson processes $X(t) = (X_1(t), \ldots, X_n(t))$, all having the same marginal rate parameter λ, that is constant over time, and coupled with each other by a given correlation structure. Each component $X_i(t) = \sum_{j=1}^{N_i(t)} \delta(t - t_j)$, where the t_j's are the time events up to t, represents a spike train, here modeled as a Poisson process. As already introduced in Chapter 1, each point process $X_i(t)$, $i = 1, \ldots, n$, can

be represented in different ways. Along with the representation as the sum of Dirac's delta functions, commonly used in neuroscience to represent spike trains, we can introduce the count process representation $N(t) = (N_1(t), \ldots, N_n(t))$. The model allows for correlations among the components by assigning to each spike in any component i a fixed probability a_k that an event occurs simultaneously to other $k-1$ other spikes, randomly sampled from $k-1$ different components $j_1, \ldots, j_k$ of the MPP, $j_l \neq i \ \forall l = 1, \ldots, k$.
Formally we can define an MPP as follows.

Definition 3.1.1. An n-dimensional stationary, homogeneous MPP with rate parameter λ and amplitude distribution $A = (a_1, \ldots, a_n)$ is a multi-dimensional process $X(t) = (X_1(t), \ldots, X_n(t))$ such that:

1. For each $i = 1, \ldots, n$, X_i is a Poisson process with stationary rate λ
2. $A = (a_1, \ldots, a_n)$ is a probability vector ($a_k \geq 0 \ \forall k$, $\sum_k a_k = 1$) and

$$\mathbb{P}(\sum_{j=1}^{n} X_j(t) = i | \sum_{j=1}^{n} X_j(t) \geq 1) =: \ a_i$$

A is called amplitude distribution. If $a_k > 0$, the MPP is said to contain correlations of order k.

Each component a_k of the amplitude distribution represents the probability that, given that one component has an event at time t, exactly $k-1$ other components contain an event at time t as well. The amplitude distribution can be defined for any multi-dimensional process and used as an alternative measure of zero-lag correlation. In our specific model the amplitude distribution represents a measure of how likely it is that different neurons synchronize their activity.
In the following we show how an MPP with given rate parameter λ and amplitude distribution $A = (a_1, \ldots, a_n)$ can be constructed by copying spikes from an "hidden" or "mother", uni-dimensional marked Poisson process into "child" processes, randomly selected for each event in the mother process.

3.1.1 Compound Poisson Process (CPP) as a generation methods for MPPs

In order to describe the technique to generate the multi-dimensional model we define first a particular marked process: the Compound Poisson Process, widely known and used in the point process literature([10, 11]).

Definition 3.1.2. A compound Poisson process (CPP) is a time-continuous doubly stochastic process with jumps. The jump times form a Poisson process and the size of the jumps is stochastic, with a specified probability distribution. A compound Poisson process, parametrized by a rate $\lambda > 0$ and jump size distribution G, is a process $\{ Y(t) : t \geq 0 \}$ given by:

$$Y(t) = \sum_{i=1}^{N(t)} D_i$$

where, $\{ N(t) : t \geq 0 \}$ is a Poisson process with rate λ, and $\{ D_i : i \geq 1 \}$ are independent and identically distributed random variables, with distribution function G, which are also independent of $\{ N(t) : t \geq 0 \}$.

In the next section we derive the generative algorithm for an MPP of rate parameter λ and amplitude distribution $A = (a_1, \ldots, a_n)$. In this procedure we need a specific CPP $N_M(t)$ that we call "mother process":

$$N_M(t) = \sum_{i=1}^{N(t)} \Xi_i \tag{3.1}$$

In order to describe entirely this compound process we have to fix the rate λ_M of the Poisson process $N(t)$ and to define the random variable Ξ from which the values Ξ_i are independently sampled.
We now define the random variable Ξ giving its discrete distribution:

$$\mathbb{P}(\Xi = k) = a_k \tag{3.2}$$

for $k = 1, \ldots, n$. This variable is entirely described by the amplitude vector A. We can compute for example its average just using the amplitude:

$$\mathbb{E}(\Xi) = \sum_{i=1}^{n} i \cdot a_i$$

Now we have to fix the rate parameter λ_M for the mother process. We choose this rate so that:

$$\mathbb{E}(N(t)) = \lambda_M t = \frac{n\lambda t}{\mathbb{E}(\Xi)}$$

Generation of a MPP

The general procedure to generate an MPP consists of the random assignment from a mother process $N_M(t)$ to each component of the multi-dimensional process $X(t)$. In particular the generative algorithm for an MPP with parameters (λ, A) is:

Algorithm 3.1.1.

1. Generate a sequence of time events $(t_1, \ldots, t_{N(T)})$ from a (mother) Poisson process $N(t)$ with rate parameter $\lambda_M = \frac{n\lambda}{\mathbb{E}(\Xi)}$ in the time interval $[0, T]$

2. Sample $N(T)$ realizations $\xi = (\xi_1, \ldots, \xi_{N(T)})$ of the random variable Ξ according to the amplitude distribution A

3. For $i = 1, ..., N(T)$:

   Extract randomly ξ_i different indices $j_1, \ldots, j_{\xi_i}$ from the set $(1, \ldots, n)$ and assign an event at time t_i to the components $X_{j_1}, \ldots, X_{j_{\xi_i}}$ of the MPP.

We need to prove that the algorithm indeed generates an MPP such that

- all components $X_i(t)$ are Poisson processes with rate parameter λ
- the amplitude of the MPP generated is A

Proposition 3.1.1.
All the components of a multi-dimensional process generated with the algorithm 3.1.1 are Poisson process with rate parameter λ.

Proof:
It is possible to notice that the inter-events intervals of all the components are exponentially distributed and independent, due to the lack of memory of the inter-events interval of the Poisson process $N(t)$. This implies that all the components are Poisson processes.

To prove that the rate of each component is λ we observe that the components generated with algorithm 3.1.1 can be described as:

$$N_j(T) = \sum_{i=1}^{N(T)} \epsilon_i \tag{3.3}$$

where ϵ_i is a binary random variable that takes a value of 1 with the probability of having assigned an event in the j-th component at the time t_i of the original Poisson process:

$$\epsilon_i = \begin{cases} 1 & \text{with } \mathbb{P}(\text{event assigned to } j\text{th margin at time } t_i) \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

Since we are randomly assigning the events between n components without repetitions, we know that:

$$\mathbb{P}(\text{event assigned to } j\text{th margin at time } t_i) = \frac{\binom{n-1}{\Xi-1}}{\binom{n}{\Xi}} = \frac{\Xi}{n}$$

To derive this equation is possible to think that $\mathbb{P}$(event assigned to j at time t_j) is equal to the probability of a hypergeometric variable of parameter (N, Ξ) is equal to 1.

Then the expected value of ϵ_i has to be:

$$\mathbb{E}(\epsilon_i) = \mathbb{E}(\frac{\Xi}{n})$$

We can now complete the proof by obtaining:

$$\mathbb{E}(N_j(T)) = \mathbb{E}(\sum_{i=1}^{N(T)} \epsilon_i) = \mathbb{E}(N(T))\mathbb{E}(\epsilon_i) =$$

$$= T\lambda_M \mathbb{E}(\frac{\Xi}{n}) = T\frac{n\lambda}{\mathbb{E}(\Xi)}\frac{\mathbb{E}(\Xi)}{n} = T\lambda$$

This implies by definition that the rate of each component N_j(t) is λ.

Proposition 3.1.2.

The multi-dimensional process generated by algorithm 3.1.1 has an amplitude distribution equal to A.

Proof:

We want to prove that the $X_j(t)$ generated with the algorithm are such that:

$$\mathbb{P}(\sum_{j=1}^{n} X_j(t) = i | \sum_{j=1}^{n} X_j(t) \geq 1) = \ a_i$$

For each time point $t_k \in (t_1, \ldots, t_{N(T)})$ in which at least one event occurs, by the construction of the algorithm the total number of synchronous events correspond to variable ξ_k. Since all the ξ_k are independently distributed as Ξ we can state that:

$$\mathbb{P}(\sum_{j=1}^{n} X_j(t) = i | \sum_{j=1}^{n} X_j(t) \geq 1) = \mathbb{P}(\xi_k = i) = \mathbb{P}((\Xi = i) = a_i$$

where the last equality is given by the definition of Ξ.

One of the useful features of the MPP model obtained from a CPP as explained above is the possibility to derive the analytical expression of many of its relevant statistical features. Below, we compute the average pairwise correlation coefficient among any pair (i, j) of components of the MPP as a function of the marginal rate λ and amplitude distribution A. It is possible to build MPPs with desired rates and pairwise correlation coefficients by inverting this procedure.

3.1.2 MPP with given average pairwise correlation coefficient

In applications of point processes, including modeling of spike train in neuroscience, pairwise correlation coefficients are often employed to quantify the strength of correlation among two processes. Pairwise correlation coefficients among spike trains can be for example computed from electrophysiological data, and then used to fit spike train models. We wish to do the same by deriving the analytical expression of the time-averaged pairwise correlation coefficient ρ_{ij} between any pair $(X_{i,}(t), X_j(t))_{i,j=1,\ldots,n}$ of components of the multivariate process, as a function of the model's rate λ and amplitude distribution A. Inverting this relation will allow us to express the amplitude distribution as a function of λ and ρ_{ij}. Because the latter two can be estimated from data, this will allow us to construct MPPs that replicate rates and pairwise correlation coefficients observed in the data. This is useful, for instance, to generate surrogate data that replicate the data's first and second moments and test for higher-order correlations.

In the remainder of this subsection we will represent all point processes as count processes, which will allow us to exploit the properties of the Poisson Process.

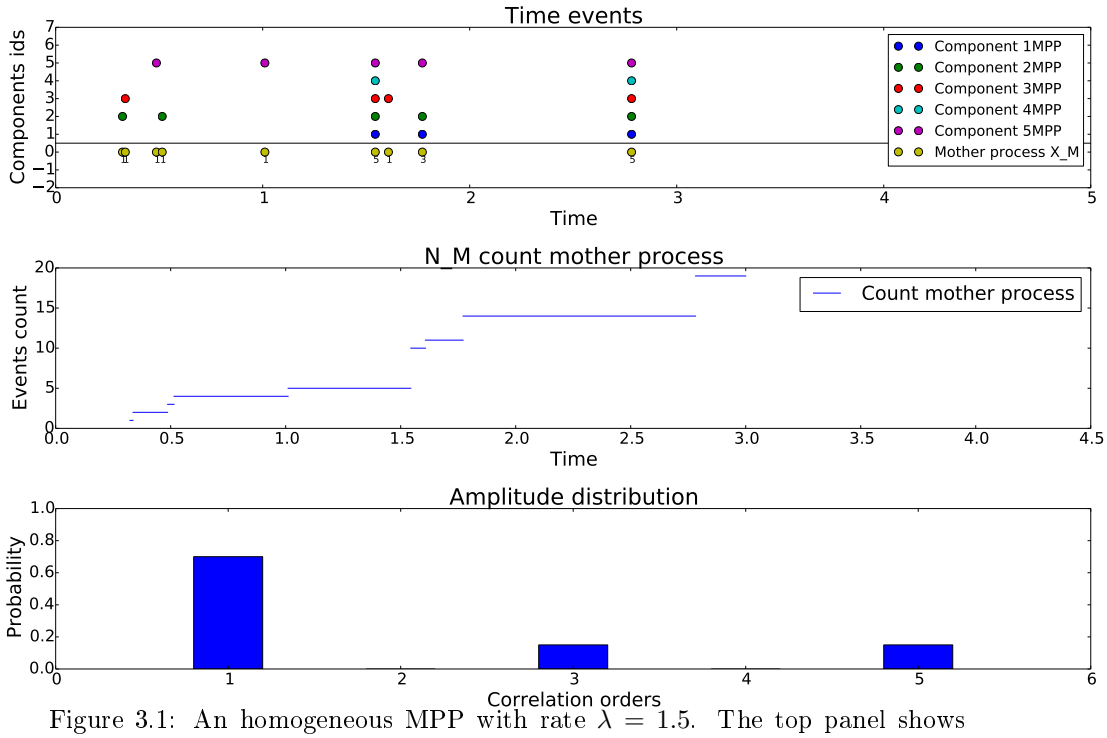

Figure 3.1: An homogeneous MPP with rate $\lambda = 1.5$. The top panel shows the time-point representation of the process and the corresponding Compound Poisson Process used as mother process in the generation process. The middle panel illustrates the count process of the total number of event. The bottom panel shows the amplitude distribution of the model with synchrony of size 3 and 5.

First, we remember the classical definition of Pearson correlation coefficient:

Definition 3.1.3. Given two random variable X and Y we define the Pearson correlation coefficient as:

$$\rho_{XY} = \frac{\mathrm{COV}(XY)}{\sqrt{\mathrm{VAR}(X)\mathrm{VAR}(Y)}}$$

We compute the average Pearson correlation coefficient across all pairs (N_i, N_j) of components of the MPP $N(t) = (N_1(t), \ldots, N_n(t))$ with parameters $(\lambda,\ A)$:

$$< \rho_{ij} > (t) = \frac{\sum_{i \neq j} p_{ij}(t)}{n(n-1)} = \frac{\sum_{i \neq j} \mathrm{COV}(X_i(t)X_j(t))}{n(n-1)\sqrt{\mathrm{VAR}(N_i(t))\mathrm{VAR}(N_j(t))}} \tag{3.5}$$

Because each component is a Poisson process with rate parameter λ, its variance is given by:

$$\mathrm{VAR}(N_i(t)) = \lambda t \quad \forall i = 1, \ldots, n \tag{3.6}$$

Then we can remember the definition of mother process and compute its variance in relation to the components and their covariance:

$$\mathrm{VAR}(N_M(t)) = \mathrm{VAR}(\sum_{i=1}^{n} N_i(t)) = \sum_{i=1}^{n} \mathrm{VAR}(N_i(t)) + \sum_{i \neq j} \mathrm{COV}(N_i(t), N_j(t)) \tag{3.7}$$

Now we can use the formulation as Compound Poisson Process of the mother count process and using its properties compute explicitly the variance:

$$\mathrm{VAR}(N_M(t)) = \mathrm{VAR}(\sum_{i=1}^{N(t)} \Xi_i) = \lambda_M t \mathbb{E}(\Xi^2) = \frac{n\lambda t}{\mathbb{E}(\Xi)} \mathbb{E}(\Xi^2) \tag{3.8}$$

Substituted the equations (3.6) and (3.7) in (3.8) it is possible to have explicitly the equation for the sum of all the covariances:

$$\sum_{i \neq j} \mathrm{COV}(N_i(t), N_j(t)) = \mathbb{E}(\Xi^2) \frac{n\lambda t}{\mathbb{E}(\Xi)} - n\lambda t \tag{3.9}$$

Finally with equations (3.6) and (3.9) we are able to derive an equation for the mean pairwise correlation coefficient depending only on the parameters (λ, A).

$$< \rho_{ij} > (t) = \frac{\mathbb{E}(\Xi^2) \frac{n\lambda t}{\mathbb{E}(\Xi)} - n\lambda t}{n(n-1)\lambda t} = \frac{\frac{\mathbb{E}(\Xi^2)}{\mathbb{E}(\Xi)} - 1}{n-1}$$

where $\mathbb{E}(\Xi) = \sum_k k \cdot a_k$ and $\mathbb{E}(\Xi^2) = \sum_k k^2 \cdot a_k$

Note that the average correlation coefficient, at least in case of homogeneous and stationary rates, is constant in time and does not depend on the rate parameter. This implies that the amplitude distribution describes completely the correlations in the process.

3.2 Single interaction process as a particular case of MPP

The MPP model introduced in Section 3.1 allows for correlations of any order among its components. This is not the case for another well established model in the computational neuroscience literature, the Single Interaction Process (SIP), which allows for correlation of maximum order only. Here we define the SIP model and we show that it represents a particular case of MIP. We also provide the relation between the two, showing how a SIP can be obtained by a suitable choice of the parameters of the MPP.

Definition 3.2.1. Single Interaction Process
The single interaction process or SIP, introduced for the first time in 2003 ([27]), is a multi-dimensional process $X(t) = (X_1(t), \ldots, X_n(t))$ in which each component X_i can be expressed as a sum of a common point process W_c and an individual contribution W_i:

$$X_i(t) = W_c(t) + W_i(t), \ \forall i = 1, \ldots, n$$

where $W_c(t)$ is a Poisson process of rate parameter λ_c and $W_i(t)$ are n i.i.d. Poisson processes with common rate parameter λ_{ind}, and independent from W_c.

Each component X_i is a Poisson process with rate parameter $\lambda_{SIP} = \lambda_c + \lambda_{ind}$ and it shares with the other components all and only the events of the process W_c. Thus, $X(t)$ has pure correlation of order n.
This process can be expressed in terms of a MPP model. Indeed, the following result holds.

Theorem 3.2.1.
A SIP model of parameter $(\lambda_{SIP}, \lambda_c)$ is equivalent to an MPP model with paramers $(\lambda_{SIP}, A_{SIP} = (a_1, 0, \ldots, 0, a_n))$ with:

$$a_1 = 1 - a_n$$

$$a_n = \frac{\lambda_c}{n\lambda_{SIP} - (n-1)\lambda_c}$$

Proof:

We prove that the amplitude distribution of a SIP model $X(t)$ is equal to A_{SIP}:

$$a_n = \mathbb{P}(\sum_{j=1}^{n} X_j(t) = n | \sum_{j=1}^{n} X_j(t) \geq 1) = \frac{\text{number of synchronous events}}{\text{synchronous events} + \text{independent events}}$$

The total number of synchronous events is given by the rate of synchrony λ_c:

$$\text{number of synchronous events} = \lambda_c T$$

The number of independent event is:

$$\text{independent events} = n\lambda_{ind} T = n(\lambda_{SIP} - \lambda_c)T$$

From this we obtain that

$$a_n = \mathbb{P}(\sum_{j=1}^{n} X_j(t) = n | \sum_{j=1}^{n} X_j(t) \geq 1) = \frac{\lambda_c}{n(\lambda_{SIP} - \lambda_c) + \lambda_c} = \frac{\lambda_c}{n\lambda_{SIP} - (n-1)\lambda_c}$$

Vice versa we can derive analytically the expected number of coincidences of an MPP with parameters $(\lambda_{SIP}, A_{SIP} = (1 - \frac{\lambda_c}{n\lambda_{SIP}-(n-1)\lambda_c}, 0, \ldots, 0, \frac{\lambda_c}{n\lambda_{SIP}-(n-1)\lambda_c}))$ and verify that it is given by the rate parameter λ_c. By the definition of amplitude distribution we can state:

$$\text{total number of synchronous events} = a_n \lambda_M T = a_n \frac{n\lambda_{SIP}}{\mathbb{E}(\Xi)} T =$$

$$= a_n \frac{n\lambda_{SIP}}{a_1 + n a_n} T = \frac{\lambda_c n \lambda_{SIP}}{(n\lambda_{SIP} - (n-1)\lambda_c)(1 - \frac{\lambda_c}{n\lambda_{SIP}-(n-1)\lambda_c} + n\frac{\lambda_c}{n\lambda_{SIP}-(n-1)\lambda_c})} T = \lambda_c T$$

3.3 MPP with heterogeneous marginal rates

We generalize here the MPP model to allow its components to have possibly different marginal rates. This will allow us in the next chapters to generate parallel spike trains with different rates, a very common scenario when dealing with electrophysiological recordings and thus an important case to address. Our goal thus becomes that of generating a MPP $X(t) = (X_1(t), \ldots, X_n(t))$ having a given amplitude distribution A and such that each component $X_i(t)$ has a (time-stationary) rate parameter λ_i.

We introduce the quantity λ_M that is as before the rate of the mother process X_M in the case of heterogeinity of the marginal rates:

$$\lambda_M = \frac{\sum_{i=1}^{n} \lambda_i}{\mathbb{E}(\Xi)}, \tag{3.10}$$

where Ξ is still defined by Equation (3.2).

In order to simulate the model we will superpose two different multi-dimensional process:

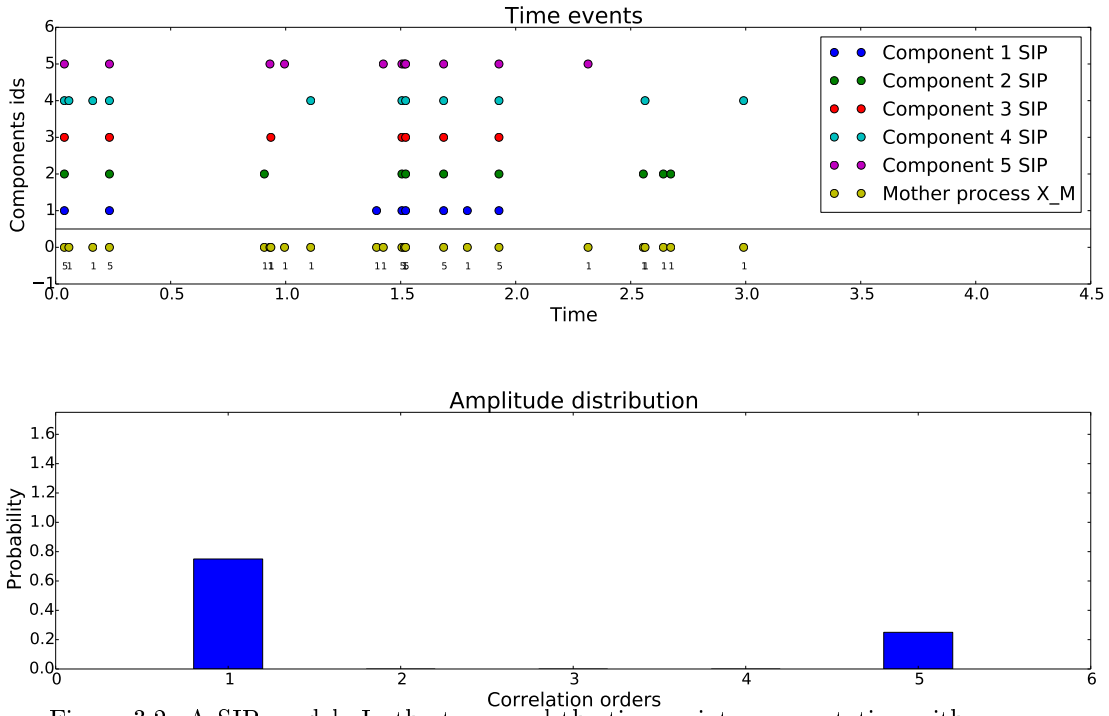

Figure 3.2: A SIP model. In the top panel the time point representation with four synchronous injections that involve all the margins. It is possible represent this model in the more general framework of MPP, in particular here it is shown the Compound Poisson Process use to generate the multi-dimensional model. In the bottom panel the two pick amplitude distribution typical of this bimodal model

- An homogeneous MPP $Z(t) = (Z_1(t), \ldots, Z_n(t))$ with correlated component and parameters (λ_Z, A_Z)
- A multi-dimensional process $I(t) = (I_1(t), \ldots, I_n(t))$ with independent components, distributed as Poisson processes with heterogeneous rates $\lambda_I = (\lambda_{I1}, \ldots, \lambda_{In})$

We set the rate parameters of this two processes:

$$\lambda_Z = \min_{i=1,\ldots,n}(\lambda_i)$$

and:

$$\lambda_{Ii} = \lambda_i - \min_{i=1,\ldots,n}(\lambda_i)$$

Then we can ulteriorly define:

$$\lambda_{M_I} = \sum_{i=1}^{n} \lambda_{Ii} = \sum_{i=1}^{n} \lambda_i - n \min_{i=1,\ldots,n}(\lambda_i) \tag{3.11}$$

and:

$$\lambda_{M_Z} = \lambda_M - \lambda_{M_I}.$$

Now in order to entirely describe the homogeneous MPP $Z(t)$ we need to fix the amplitude vector A_Z:

$$\begin{cases} a_{Zi} = \frac{\lambda_M}{\lambda_{M_Z}} a_i & for\ i \neq 1 \\ a_{Z1} = \frac{\lambda_M}{\lambda_{M_Z}} a_1 - \frac{\lambda_{M_I}}{\lambda_{M_Z}} \end{cases} \tag{3.12}$$

A further check to be carried out is that the amplitude A_Z is indeed a probability vector, that is, $a_{Zi} \geq 0\ \forall i$ and $\sum_{i=1}^{n} a_{Zi} = 1$.

The second condition holds true, because:

$$\sum_{i=1}^{n} a_{Zi} = \frac{\lambda_M}{\lambda_{M_Z}} a_1 - \frac{\lambda_{M_I}}{\lambda_{M_Z}} + \sum_{i=2}^{n} (\frac{\lambda_M}{\lambda_{M_Z}} a_i) = \frac{\lambda_M}{\lambda_{M_Z}} \sum_{i=1}^{n} a_i - \frac{\lambda_{M_I}}{\lambda_{M_Z}} = \frac{\lambda_M - \lambda_{M_I}}{\lambda_{M_Z}} = 1$$

The first condition is also always satisfied for a_{Zi} with $i = 2, \ldots, n$, since all elements of the product are positive. For the remaining case $i = 1$ it is possible to deduce a condition on the parameter of the output process $X(T)$ such that:

$$a_{Z1} = \frac{\lambda_M}{\lambda_{M_Z}} a_1 - \frac{\lambda_{M_I}}{\lambda_{M_Z}} \geq 0$$

Which implies:

$$\frac{1}{\lambda_{M_Z}} (\lambda_M a_1 - \lambda_{M_I}) \geq 0$$

Since $\frac{1}{\lambda_{M_Z}}$ is always positive we can derive the analytical constrain for the parameter of a heterogeneous MPP:

$$a_1 - \frac{\lambda_{M_I}}{\lambda_M} \geq 0$$

It is easy to think at many configurations in the parameter space that are not solvable (e.g. any choose of heterogeneous rates with an amplitude in form $A = (0, \ldots, 0, 1)$ that correspond to complete synchrony of all the components, that of course it's a contradiction with the assumption of different rate in different component).

We can proceed presenting the algorithm for generation of a MPP model with heterogeneous rate parameters $\lambda = (\lambda_1, \ldots, \lambda_n)$ and amplitude A in a time interval $[0, T]$:

Algorithm 3.3.1.

1. Generate a multi-dimensional Poisson process with independent components each with rate parameter $(\lambda_i - \min_{i=1,\ldots,n}(\lambda_i))$
2. Generate a multi-dimensional homogeneous MPP with rate parameter:

   $$(\min_{i=1,\ldots,n}(\lambda_i))$$

   and amplitude distribution given by equation (3.12)
3. Superpose the two multi-dimensional process component by component obtaining a new multi-dimensional process $X(t)$

Now we show tat the multi-dimensional process generated is the desisred MPP. We start to prove that alle the components are Poisson process with the desired rates.

Proposition 3.3.1.
The components of the process generated via the algorithm 3.3 are Poisson-distributed, and the rate parameter of component X_i is λ_i.

Proof:
Each component of the output process is generated from the superposition:

$$X_i(t) = Z_i(t) + I_i(t)$$

Since both $Z_i(t)$ and $I_i(t)$ are independent Poisson process the sum is a Poisson process with rate parameter given by:

$$\lambda_Z + \lambda_{Ii} = \min_{i=1,\ldots,n}(\lambda_i) + \lambda_i - \min_{i=1,\ldots,n}(\lambda_i) = \lambda_i$$

Then we check for the amplitude distribution.

Proposition 3.3.2.
The amplitude distribution of the multi-dimensional process generate with algorithm (3.3) is equal to A

Proof:
As already done in previous demonstration we can use that each entries of the amplitude is given by:

$$\mathbb{P}(\sum_{j=1}^{n} X_j(t) = i | \sum_{j=1}^{n} X_j(t) \geq 1) = \frac{\text{number of synchronous events of size i}}{\text{synchronous events of all sizes} + \text{independent events}}$$

The sum of the number of events is given by the sum of the events of the superposed processes:

$$\text{synchronous events of all sizes+independent events} = \sum_{i=2}^{n} a_i \lambda_{M_Z} + a_{Z1}\lambda_{M_Z} + \lambda_{M_I} =$$

$$= \lambda_{M_Z} + \lambda_{M_I} = \lambda_M$$

We split now the argumentation in the two cases for $i = 1$ and $i = 2, \ldots, n$.
In the case of $i = 1$ the number of events correspond to the number of independent events given by:

$$\text{independent events} = \lambda_{M_I} + a_{Z1}\lambda_{M_z} = \lambda_{M_I} + (\frac{\lambda_M}{\lambda_{M_Z}} a_1 - \frac{\lambda_{M_I}}{\lambda_{M_Z}})\lambda_{M_z} =$$

$$= \lambda_{M_I} + \lambda_M a_1 - \lambda_{M_I} = \lambda_M a_1$$

Then we can derive:

$$\mathbb{P}(\sum_{j=1}^{n} X_j(t) = 1 | \sum_{j=1}^{n} X_j(t) \geq 1) = \frac{\text{number of independent events}}{\text{synchronous events of all sizes} + \text{independent events}} =$$

$$= \frac{\lambda_M a_1}{\lambda_M} = a_1$$

For the case of synchrony of order $i = 2, \ldots, n$, the total number of synchrony of size i is given by:

$$\text{number of synchronous events of size i} = a_{Zi}\lambda_{M_z} =$$

$$= \frac{\lambda_M}{\lambda_{M_Z}} a_i \lambda_{M_Z} = \lambda_M a_i$$

Hence the ith entry of the amplitude distribution is:

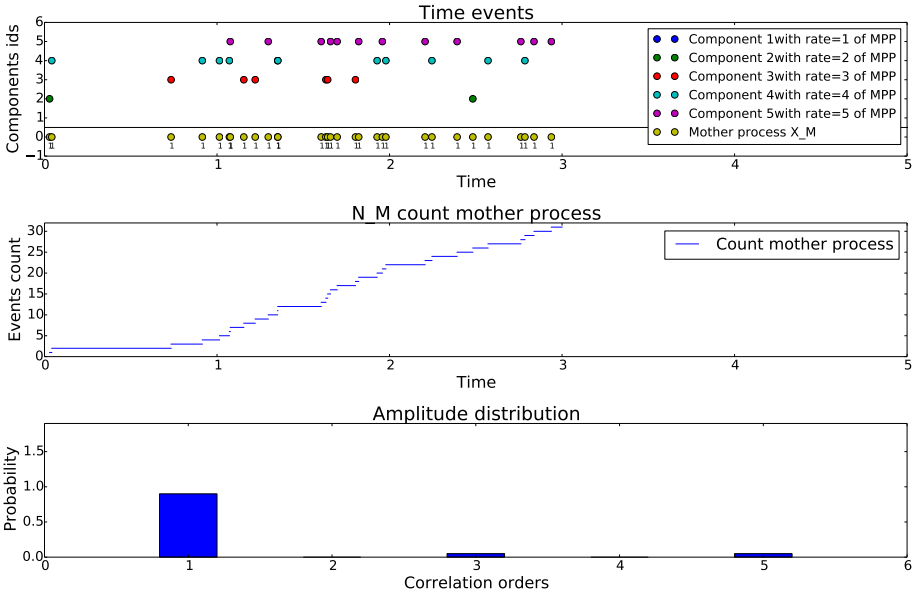

Figure 3.3: An heterogeneous MPP with rate $\lambda = (1, 2, 3, 4, 5)$. On top panel its time-point representation and the correspondent Compound Poisson Process used as mother process in the generation process. In the middle panel the count process of the total number of event. In the bottom panel the amplitude distribution of the model presenting synchrony of size 3 and 5

$$\mathbb{P}(\sum_{j=1}^{n} X_j(t) = i | \sum_{j=1}^{n} X_j(t) \geq 1) = \frac{\text{number of synchronous events of size i}}{\text{synchronous events of all sizes} + \text{independent events}} =$$

$$\frac{\lambda_M a_i}{\lambda_M} = a_i$$

With this we have proved that the amplitude distribution of the output process is the probability vector A.

3.4 MPP with non-stationary rates

A necessary extension to make the model more flexible is to move to non-stationary Poisson processes. In this framework the constant rate parameter it is substituted with a function of time $\lambda(t)$.

In this context the output process that we want to describe is a multi-dimensional process $X(t)$ with the same rate profile $\lambda(t)$ for all the component

and the correlation structure represented by the amplitude distribution A defined as before. By definition of rate profile we know that:

$$\mathbb{E}(N_i(t)) = \int_0^t \lambda(s)ds$$

As for the previous cases we can introduce the mother process. Now it consists of a a non-stationary CPP with rate profile:

$$\lambda_M(t) = \frac{n\lambda(t)}{\mathbb{E}(\Xi)}$$

The algorithm to generate a MPP process with homogeneous marginal rate profile $\lambda(t)$ and amplitude distribution A is:

Algorithm 3.4.1.

1. Generate a non-stationary Poisson process $N(t)$ with parameter $\lambda_M(t)$ equal to $\frac{n\lambda(t)}{\mathbb{E}(\Xi)}$ for t in the time interval $[0, T]$
2. Sample from the distribution A $N(T)$ realization of the random variable Ξ: $\xi = (\xi_1, \ldots, \xi_{N(T)})$
3. For $i = 1, ..., N(T)$:

   Extract randomly ξ_i different indexes $j_1, \ldots, j_{\xi_i}$ from the set $(1, \ldots, n)$ assign an event at t_i, time of the i-th event of the Poisson process $N(t)$, at the components $X_{j_1}, \ldots, X_{j_{\xi_i}}$ of the output process

As like as for the stationary case we prove that the processes generated with this algorithm is the desired MPP.

Proposition 3.4.1.

All the components of the multi-dimensional process generated with algorithm 3.4 are non-stationary Poisson porcesses with rate profile $\lambda(t)$.

Proof:

All the components generated are Poissonian, because partition of the Poisson process $N(t)$.

Then we just need to prove that the rate profile of each component is the desired $\lambda(t)$. We can proceed as in the proof of 3.1.1. We obtain the non-stationary version for 3.3:

$$\mathbb{E}(N_j(T)) = \mathbb{E}(\sum_{i=1}^{N(T)} \epsilon_i) = \mathbb{E}(N(T))\mathbb{E}(\epsilon_i) = \int_0^T \lambda_M(s)ds\mathbb{E}(\frac{\Xi}{n}) = \int_0^T \frac{n\lambda(s)}{\mathbb{E}(\Xi)}ds\frac{\mathbb{E}(\Xi)}{n} = \int_0^T \lambda(s)ds$$

where ϵ_i are defined as in 3.4.

Proposition 3.4.2.
The multi-dimensional process generated with algorithm 3.4 has an amplitude distribution equal to A.

Proof:
For the proof we refer to proposition 3.1.1. Since the rate does not influence the amplitude the amplitude distribution the proof are equvalent for the non-stationary and stationary cases.

It is possible to find in literature other frameworks to generate an MPP model with non-stationary rate, in particular in [37]. The advantage of our algorithm is the variety of the amplitude distributions that is possible to obtain. Indeed using the framework suggested in [37] it is possible construct correlated multi-dimensional general point process (both non-stationary and non poissonian) but only for specific final amplitude distribution. In particular the use of thinning method on single components constrain the amplitude distribution to have specific properties (e.g. with this method it is not possible to extend the SIP process to the more general MPP for the non-stationary version, but with our implementation is the natural extension of Section 3.2 to the non-stationary case).

Another problematic that in [37] is not treated explicitly the case of different rate profiles for the different components. In the next section we show how it is possible extend our framework to a general context of heterogeneous non-stationary rates.

3.5 MPP with heterogeneous non-stationary rates

The successive step to enlarge the adaptability of the model is to generate a multi-dimensional process $X(t)$ in which all the components may have different time depending rate profiles. The final components $X_i(t)$ are correlated Poisson processes each with rate profile$\lambda_i(t)$ for $i = 1, \ldots, n$.

As for the stationary case in the generative algorithm we use two processes to be superposed:

- An homogeneous MPP $Z(t) = (Z_1(t), \ldots, Z_n(t))$ with correlated components (amplitude distribution A_Z) and non-stationary rate parameters $\lambda_Z(t)$
- A multi-dimensional process $I(t) = (I_1(t), \ldots, I_n(t))$ with independent components, distributed as Poisson processes with heterogeneous non-stationary rates $\lambda_I = (\lambda_{I1,}(t) \ldots, \lambda_{In}(t))$

We set the rate for the MPP $Z(T)$ taking the minimum rate in each time point:

$$\lambda_Z(t) := \min_{i=1,\ldots,n} (\lambda_i(t))$$

Consequently we define the rate profile for each of the independent processes:

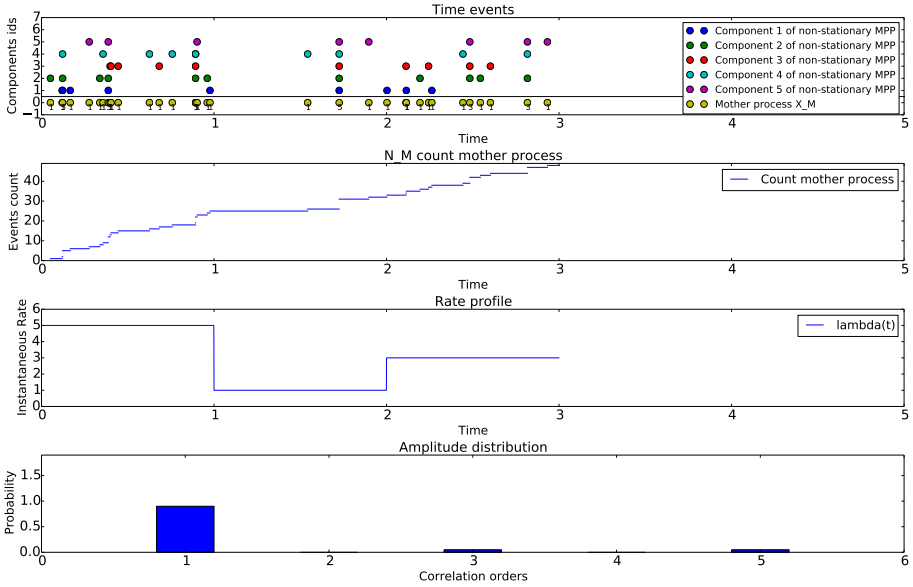

Figure 3.4: An non-stationary MPP with homogeneous rate function $\lambda(t) = 5$ for $t \in (0, 1]$, $\lambda(t) = 1$ for $t \in (1, 2]$, $\lambda(t) = 3$ for $t \in (2, 3]$. On top panel its time-point representation and the correspondent Compound Poisson Process used as mother process in the generation process. In the middle panel the count process of the total number of event and the rate profile of each components. In the bottom panel the amplitude distribution of the model presenting synchrony of syze 3 and 5

$$\lambda_{Ii}(t) = \lambda_i(t) - \min_{i=1,\dots,n}(\lambda_i(t))$$

We can compute the rate profile of the non-stationary CPP that is the mother process of the MPP:

$$\lambda_M(t) = \frac{\sum_{i=1}^{n}\lambda_i(t)}{\mathbb{E}(\Xi)}$$

We need also the quantities given by:

$$\lambda_{M_I}(t) = \sum_{i=1}^{n}\lambda_{Ii}(t)$$

and:

$$\lambda_{M_Z}(t) = \lambda_M(t) - \lambda_{M_I}(t)$$

Now in order to entirely describe the homogeneous $Z(t)$ we define its amplitude distribution A_Z as:

$$\begin{cases} a_{Zi} = \frac{\int_0^T \lambda_M(s)ds}{\int_0^T \lambda_{M_Z}(s)ds} a_i & for\ i \neq 1 \\ a_{Z1} = \frac{\int_0^T \lambda_M(s)ds}{\int_0^T \lambda_{M_Z}(s)ds} a_1 - \frac{\int_0^T \lambda_{M_I}(s)ds}{\int_0^T \lambda_{M_Z}(s)ds} \end{cases} \tag{3.13}$$

In order to check that A_Z it is a probability vector we compute:

$$\sum_{i=1}^{n} a_i = \frac{\int_0^T \lambda_M(s)ds}{\int_0^T \lambda_{M_Z}(s)ds}\sum_{i=2}^{n} a_i + \frac{\int_0^T \lambda_M(s)ds}{\int_0^T \lambda_{M_Z}(s)ds}a_1 - \frac{\int_0^T \lambda_{M_I}(s)ds}{\int_0^T \lambda_{M_Z}(s)ds} =$$

$$\frac{\int_0^T \lambda_M(s)ds}{\int_0^T \lambda_{M_Z}(s)ds} - \frac{\int_0^T \lambda_{M_I}(s)ds}{\int_0^T \lambda_{M_Z}(s)ds} = \frac{\int_0^T \lambda_M(s) + \lambda_{M_I}(s)ds}{\int_0^T \lambda_{M_Z}(s)ds} = 1$$

We need also that the following condition is satisfied for $i = 1, \dots, n$:

$$a_{Zi} \geq 0$$

In the cases with $i = 2, \dots, n$ this condition is satisfied, since all the elements of the product are positive by definition.

For $i = 1$ we can derive the condition on the parameter of the heterogeneous non-stationary MPP to be generated:

$$a_{Z1} = \frac{\int_0^T \lambda_M(s)ds}{\int_0^T \lambda_{M_Z}(s)ds}a_1 - \frac{\int_0^T \lambda_{M_I}(s)ds}{\int_0^T \lambda_{M_Z}(s)ds} \geq 0$$

$$\frac{1}{\int_0^T \lambda_{M_Z}(s)ds}(\int_0^T \lambda_M(s)dsa_1 - \int_0^T \lambda_{M_I}(s)ds) \geq 0$$

$$a_1 - \frac{\int_0^T \lambda_{M_I}(s)ds}{\int_0^T \lambda_M(s)ds} \geq 0$$

This is the condition that the rates and the amplitude of the process has to satisfy to be generated with the following algorithm.

We can now formulate the algorithm for the generation of an heterogeneous non-stationary MPP:

Algorithm 3.5.1.

1. Generate a multi-dimensional Poisson process with independent components each with rate profiles $\lambda_i(t) - \min_{i=1,\dots,n}(\lambda_i(t))$
2. Generate a multi-dimensional homogeneous MPP with rate function $\lambda_Z(t) = \min_{i=1,\dots,n}(\lambda_i(t))$ and amplitude distribution given by equation (3.13)
3. Superpose the two multi-dimensional processes component by component obtaining a new multi-dimensional process $X(t)$

We can prove that the multi-dimensional so generated is the desired MPP.

Proposition 3.5.1.

Each component $X_i(t)$ of the multi-dimensional process generated with algorithm 3.5 are Poisson process with rate parameter $\lambda_i(t)$.

Proof:

Each component is a Poisson process because it is the superposition of two Poisson process $X_i(t) = Z_i(t) + I_i(t)$.

Then we can compute the rate parameter of each component:

$$\mathbb{E}(\int_0^t X_i(s)ds) = \mathbb{E}(\int_0^t Z_i(s) + I_i(s)ds) = \int_0^t \lambda_Z(s) + \lambda_{Ii}(s)ds =$$

$$= \int_0^t \min_{i=1,\dots,n}(\lambda_i(s)) + \lambda_i(s) - \min_{i=1,\dots,n}(\lambda_i(s))ds = \int_0^t \lambda_i(s)$$

Proposition 3.5.2.

The amplitude of the multi-dimensional process generated with algorithm 3.5 is the probability vector A

Proof:

As already done in previous demonstrations we can use that each entries of the amplitude is given by:

$$\mathbb{P}(\sum_{j=1}^n X_j(t) = i | \sum_{j=1}^n X_j(t) \geq 1) = \frac{\text{number of synchronous events of size i}}{\text{synchronous events of all sizes} + \text{independent events}}$$

The sum of the number of events is given by the sum of the events of the superposed processes:

$$\text{synchronous events of all sizes} + \text{independent events} =$$

$$= \sum_{i=2}^{n} a_i \int_0^T \lambda_{M_Z}(s)ds + a_{Z1} \int_0^T \lambda_{M_Z}(s)ds + \int_0^T \lambda_{M_I}(s)ds =$$

$$= \int_0^T \lambda_{M_Z}(s) + \lambda_{M_I}(s)ds = \int_0^T \lambda_M$$

We split now the argumentation in the two cases for $i = 1$ and $i = 2, \ldots, n$.

In the case of $i = 1$ the number of events correspond to the number of independent events given by:

$$\text{independent events} = \int_0^T \lambda_{M_I}(s)ds + a_{Z1} \int_0^T \lambda_{M_Z}(s)ds =$$

$$= \int_0^T \lambda_{M_I}(s)ds + \int_0^T (\frac{\lambda_M(s)}{\lambda_{M_Z}(s)} a_1 - \frac{\lambda_{M_I}(s)}{\lambda_{M_Z}(s)}) \lambda_{M_z}(s)ds =$$

$$= \int_0^T \lambda_{M_I}(s)ds + a_1 \int_0^T \lambda_M(s)ds_1 - \int_0^T \lambda_{M_I}(s)ds = a_1 \int_0^T \lambda_M(s)ds$$

Then we can derive:

$$\mathbb{P}(\sum_{j=1}^{n} X_j(t) = 1 | \sum_{j=1}^{n} X_j(t) \geq 1) = \frac{\text{number of independent events}}{\text{synchronous events of all sizes} + \text{independent events}} =$$

$$= \frac{a_1 \int_0^T \lambda_M(s)ds}{\int_0^T \lambda_M(s)ds} = a_1$$

For the case of sinchrony of order $i = 2, \ldots, n$, the total number of synchrony of size i is given by:

$$\text{number of synchronous events of size i} = a_{Zi} \int_0^T \lambda_{M_Z}(s)ds =$$

$$= \int_0^T \frac{\lambda_M(s)}{\lambda_{M_Z}(s)} a_i \lambda_{M_Z}(s)ds = \int_0^T \lambda_M(s)ds a_i$$

Hence the ith entry of the amplitude distribution is:

$$\mathbb{P}(\sum_{j=1}^{n} X_j(t) = i | \sum_{j=1}^{n} X_j(t) \geq 1) = \frac{\text{number of synchronous events of size i}}{\text{synchronous events of all sizes} + \text{independent events}} =$$

$$\frac{\int_0^T \lambda_M(s)dsa_i}{\int_0^T \lambda_M(s)ds} = a_i$$

With this we have proved that the amplitude distribution of the output process is the probability vector A.

We have now available a correlated multi-dimensional model with a very adaptable structure that can fit all the possible rate configuration under the hypothesis of Poisson spike trains.

In the next section we introduce the statistical methods for high order correlation detection that we will apply in a comparative fashion to data generated according the MPP model (Chapter 5).

Chapter 4

Statistical methods for HOC detection

In this chapter we will introduce the statistical methods that we will use for our comparative analysis in the final chapter. These are methods specifically developed for the detection of higher orders correlations in massively parallel spike trains.

The first method we will consider is the Cumulant-Based Inference of Correlation, or CuBIC [42]. This methods aims at determining the minimum order of correlation necessary to explain observed synchrony in the data at the population level. It does so by successive statistical tests that assess any given order of correlation assumed correlations of lower order. The result is a positive integer ξ that represents the lowest size of significant (i.e. non-chance) synchronous events to be assumed in order to explain the observed population synchrony.

The second method under consideration exploits data-mining techniques to efficiently search in parallel spike trains for repeating patterns of synchronous spikes, and then assesses their statistical significance under the null hypothesis of independence through Monte-Carlo simulations ([43]). Despite this method lacking a name at the time of publication, following the authors' suggestion, we will refer to it as the Spike Pattern Detection and Evaluation (SPADE) analysis. SPADE determines specific patterns of synchronous spikes which are considered statistically unexpected under independence. Our goal here is to investigate the relation between the two methods by comparing their outcome on identical data sets with various statistical features.

Before detailing the methods, we need to define spike synchrony, on which the next section concentrates on.

4.1 Binning procedure

Spike trains as defined in Chapter 2 are time-continuous point processes. Considered the biological nature of the underlying generating processes, synchrony

among spike trains is not expected to be infinitely precise. In mathematical terms, the point processes used to represent parallel spike trains (as illustrated in Chapter 2) have zero probability to show infinitely precise correlations. Thus, the question of whether or not neurons exhibit synchronous firing must be understood as asking whether or not they exhibit imprecise synchrony. Typically one allows for some time jitter (a few milliseconds) among spikes that one wishes to still define as synchronous. Spikes beyond this imprecision level are then considered as not synchronous. A common approach to determine synchrony consists in discretizing the time axis into exclusive bins of a few ms, and then to transform each time-continuous spike train into a discrete time series of zeroes (no spike in the bin) and ones (one or more spikes in the bin). This procedure, called spike train binning or time binning, is rephrased below in mathematical terms.

Given a realization of a general point process $X(t)$, or in particular a spike train, in a time interval $(t_{start}, t_{stop}]$ it is possible to derive a representation with a discretized time axis. We fixed a time interval length h that divide the interval length $(t_{stop} - t_{start})$ and that is the unit of our discretization. We construct a partition, or binning, of the interval with the set of sub-intervals, or bins, defined:

$$\Delta t_i = (t_{start} + (i-1)h, t_{start} + ih]$$

for $i = 1, \ldots, \frac{t_{stop}-t_{start}}{h}$. Then we can describe the point process as a series $(x(\Delta t_i))_{i=1}^{\frac{t_{stop}-t_{start}}{h}}$. Every element of the succession assumes a value according:

$$x(\Delta t_i) = \int_{\Delta t_i} X(s)ds = N(t_{start} + ih) - N(t_{start} + (i-1)h)$$

This representation consists in the count of events, spikes, per bin.

In most occasions it is desirable to consider the so called clipped version of this series. It consists of a binary series and it is defined as:

$$x(\Delta t_i) = \begin{cases} 1 & \text{if } \exists t \in \Delta t_i | X(t) = 1 \\ 0 & \text{otherwise} \end{cases}$$

This representation consists of assigning the value 1 to all the bins of the discretization in which at least one of the events of the process $X(t)$ occur, 0 otherwise.

In particular we will show that the two methods that we will consider in this discussion work with discretized, binned version of the data. It is possible to derive a formal relation between the original data and the discretized version:

$$\lim_{h \to 0}(x_i) = X(t)$$

This relation permits to extend many of the statistical properties of the time series to the binned version.

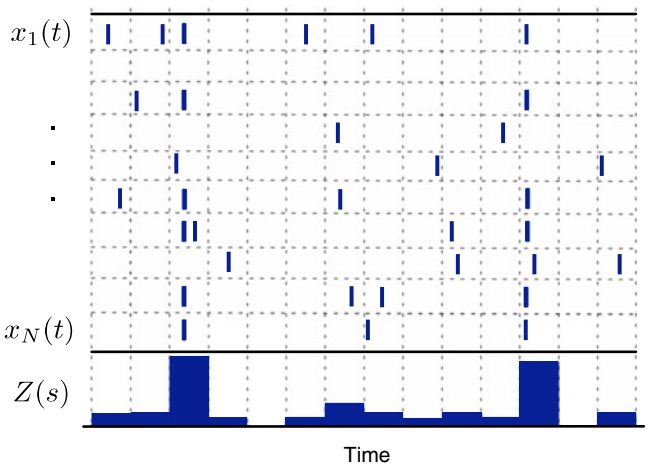

Figure 4.1: Binned realization of a multi-dimensional point process and the related population histogram (modified from [42])

In the context of a multi-dimensional point processes it is possible to proceed equivalently to the discretization of the time axes, operating components per components. It is possible to use different bin sizes for different components. In our work we will always consider the length of bin size of discretization constant, since we want to compare the different components on the same temporal scale, because we are interested in synchronous events.

After creation of the binned version of each single spike train we introduce a new object: the count of events per bin across the time series, called population histogram (Figure 4.1).

Definition 4.1.1.

Given a multi-dimensional process $X(t) = (X_1(t), \ldots, X_n(t))$ and its discretized version of each component $x_j(\Delta t_i) = (x_j(\Delta t_i))_{i=1}^{\frac{t_{stop}-t_{start}}{h}}$, the population histogram is defined in the bin Δt_i as:

$$z(\Delta t_i) = \sum_{j=1}^{n} x_j(\Delta t_i)$$

This quantity can be computed both for the unclipped or clipped time series. Using the binary binning it assumes values between 0 and n because it is the sum of n binary elements. It can also be considered as the representation of the complexity of the synchrony with a time precision equal to h.

4.2 Surrogates

In time series analysis a technique largely used to implement a particular null hypothesis, in particular for correlation analysis, is to generate artificial data using a transformation of the original data set. The main idea behind this technique is to create data that preserve some of the statistical features of the original data set while destroying others, in particulare the ones we aim to test for. Data that are generated this way are commonly called surrogate data. Surrogates can be obtained either from the original spike trains or from their binned version.

Many different types of surrogates have been proposed, able to retain and destroy different features of the original data at different degrees ([28], [19]). Here we introduce in detail the two surrogates that we will use when we apply the SPADE analysis in chapter 5.

There exists further methods, which have different properties(Figure 4.2). Some other examples are reported in Chapter 17 of [19].

4.2.1 Event (Spike) dithering

Spike dithering consists of the displacement of each single time events (spike) (Figure 4.2 panel d).

Definition 4.2.1.

Given a point process $X(t)$ and a specific realization $(t_1, \ldots, t_n)$ the dithered surrogate $S(t)$ of dither parameter $\Delta \in \mathbb{R}^+$ is defined as:

$$S(t) = \sum_{i=0}^{n} \delta(t - (t_i + u_i))$$

where the u_is are independent random variables uniformly distributed in $[-\Delta, \Delta]$.

The parameter Δ of the technique determines statistically the difference between the surrogate and the original data. Optimal values for Δ should be chosen in such a way to preserve marginal statistics of the individual neurons, such as total spike counts and firing rates computed on a time scale larger then Δ while destroying synchronous events of the original data.

4.2.2 Spike train shifting

The second technique that we will threat is the time shifting. This framework does not consist in the displacement of each single event independently, but a single uniform random variable is generated for each point process, or spike train.

Definition 4.2.2.

Given a point process $X(t)$ the shifted surrogate process $S(t)$ of shifting parameter $\Delta \in \mathbb{R}^+$ is defined as:

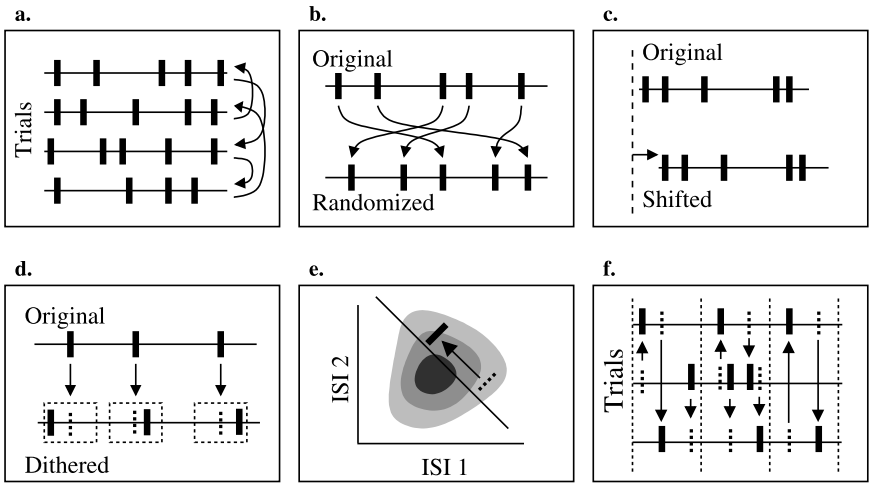

Figure 4.2: Surrogates: figure from [19] with six different procedure to generate surrogate data

$$S(t+u) = X(t)$$

where u is a random variable uniformly distributed in the interval $[-\Delta, \Delta]$.

This procedure (Figure 4.2 panel c), exactly as for the spike dithering, generates surrogates so that the synchrony are destroyed. Indeed each uniform variable is generated independently for each margins.

Differently from the previous example the surrogate generated have exactly the same inter event intervals. Indeed the original inter-event intervals ΔX_i are preserved:

$$\Delta_i X := t_i - t_{i-1} = t_i + u - t_{i-1} - u =: \Delta_i S$$

4.3 CuBIC

The Cumulant-Based Inference of higher-Correlation is a statistical method for the detection of high order correlation in multivariate point processes. It has been developed [42] in the context of electrophysiological data analysis but it can be extended to any kind o multi-dimensional point process. As we show in details it is based on the assumption of a MPP model. The method is finalized to infer the minimum order of correlation necessary to explain the dataset analyzed, assuming the underlying model of a MPP process.

4.3.1 Binary representation of the MPP model

Before to present the details of the methods we have to start to explain in detail how the original continuous data, formed by time point, are elaborated. Indeed the method is constructed for the analysis of the binned version of the data. This means that the time event series is discretized as explained in Section 4.1 fixing a certain time unit h.

Since the method is constructed on the MPP model it is useful to specify some characteristic of the discretized version of this model and to introduce some notations for the binned version of the MPP.

From now on we will consider a MPP process $X(t) = (X_1(t), \dots, X_n(t))$ with homogeneous stationary rate parameter λ and amplitude distribution A.

In this context we consider the discretized version of each margins in the interval $(t_{start}, t_{stop}]$ fixing a bin size h, in particular we will take into account the clipped binning.

We construct the partition $\Delta t_i = (t_{start} + (i-1)h, t_{start} + ih]$ for $i = 1, \dots, T = \frac{t_{stop} - t_{start}}{h}$.

We can successively consider a vector of n binary series $(x(\Delta t_i))_{i=1}^T = (x_1(\Delta_i), \dots, x_n(\Delta_i))_{i=1}^T$ as the binned version of the MPP $X(t)$.

Since all the margins $X_i(t)$ are marginally Poisson process and we fix $h << 1$ we can also assume that:

$$\mathbb{E}(x_j(\Delta_i) = 1) = \lambda h$$

for all $i = 1, \dots, T$ and $j = 1, \dots, n$.

The same discretization can be considered for the correspondent mother process X_M related to the MPP. In this case we will not take into account the binary version of the process but the total count per bin. Thins means that the discretized version in a particular bin Δt_i is defined as:

$$x_M(\Delta t_i) = \int_{\Delta t_i} X_M(s) = N_M(t_{start} + ih) - N_M(t_{start} + (i-1)h) \tag{4.1}$$

Proposition 4.3.1.

The binned version of the mother process of a MPP $X(t)$ is equivalent to the population histogram (section 4.1).

Proof:

By construction of MPP (section 3.1) the mother process is:

$$X_M(t) = \sum_{j=1}^{n} X_j(t)$$

Applying the 4.1 to get the binned sequence we obtain:

$$x_M(\Delta t_i) = \int_{\Delta t_i} X_M(s)ds = \int_{\Delta t_i} \sum_{j=1}^{n} X_j(s)ds = \sum_{j=1}^{n} \int_{\Delta t_i} X_j(s)ds = \sum_{j=1}^{n} x_j(\Delta t_i)$$

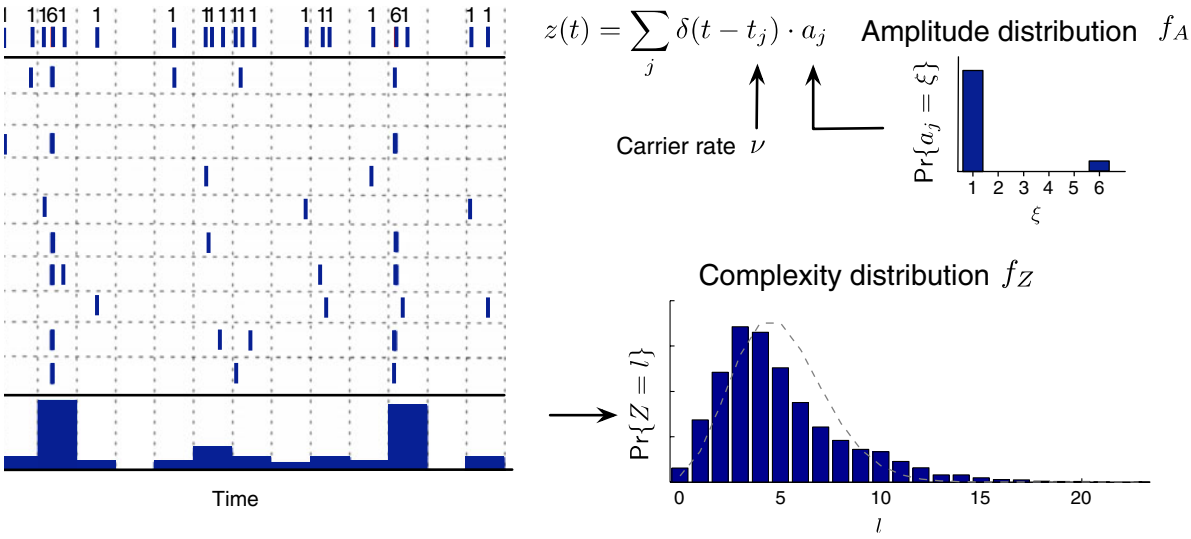

Figure 4.3: Binned representation of a MPP with the relative amplitude and complexity distributions (modified from [42])

This is equivalent to the population histogram of $X(t)$ as defined in (4.1).

Because of our initial assumption of stationarity the distribution of the count of event $x_M(\Delta t_i)$ does not depend on time.

In particular we can think to introduce a random variable Z, with distribution given by:

$$\mathbb{P}(Z = z) = \mathbb{P}(x_M(\Delta t_i) = z) \tag{4.2}$$

We will call the distribution of Z complexity distribution. The complexity distribution is strictly connected with the correlation of the multi-dimensional process, indeed it corresponds to the probability to have exactly z synchronous events in a certain time window (bin) h.

In the next section we will show how this distribution, in particular its cumulants, that can be estimated from the data, can be used to construct hypothesis tests to infer the order of correlation of the data.

4.3.2 Cumulants

In this section we will analytically derive the generic cumulant ([29]) of order m of the random variable Z for a MPP model with parameter (λ, A).

Theorem 4.3.1.

Given an MPP process with parameter (λ, A) with the relative mother process X_M, the m-th cumulant of the random variable defined in equation (4.2) is given by:

$$k_m(Z) = \mu_m(\Xi)\lambda_M h$$

where λ_M is the rate parameter of the mother process and $\mu_m(\Xi)$ is the mth moment of the random variable Ξ defined in equation (3.2).

Proof:

We define the random variables Y_l for $l = 1, \ldots, n$ as the number of synchronous events of size l per time bin.

This variables are a partition of the Poisson variable $\Delta N(h) := N(t) - N(t+h) \sim P(\lambda h)$, where $N(t)$ is the Poisson process used to define the mother process in equation (3.1). In particular each event of $\Delta N(h)$ is counted for Y_l with probability equal to a_l. Using the result in Prop. 1.1.2 we can state that:

$$Y_l \sim P(\lambda_l = a_l \lambda_M h)$$

We write now the variable Z as $Z = \sum_{l=1}^{n} lY_l$.

We can use use this equation to explicitly derive the cumulants of the variable Z. We apply the scalar properties of cumulants k for which:

$$k_m(cX) = c^m k_m(X)$$

if c is a constant.

We also know that, since Y_m is a Poisson random variable of parameter λ_l, $k_m(Y_l) = \lambda_l$.

We have all the elements to compute the cumulant of order m of Z:

$$k_m(Z) = k_m(\sum_{l=1}^{n} lY_l) = \sum_{l=1}^{n} l^m k_m(Y_l) = \sum_{l=1}^{n} l^m \lambda_l = \sum_{l=1}^{n} l^m a_l \lambda_M h$$

Now we can remember that in chapter 3 we had introduced the random variable Ξ. This is a discrete variable distributed according to the amplitude distribution A. Its moments are defined as:

$$\mu_m(\Xi) = \mathbb{E}(\Xi^m) = \sum_{l=1}^{n} l^m \mathbb{P}(\Xi = l) = \sum_{l=1}^{n} l^m a_l$$

We can now find the explicit relation between the random variable Z (in particular its cumulants) and the amplitude distribution:

$$k_m(Z) = \mu_m(\Xi)\lambda_M h \tag{4.3}$$

For the construction of the MPP model we already know how the amplitude is strictly related to the correlations between the margins. In particular we know that the maximum order of correlation correspond to the value $\xi = \text{argmax}_{i=1,\ldots,n}(a_i \neq 0)$.

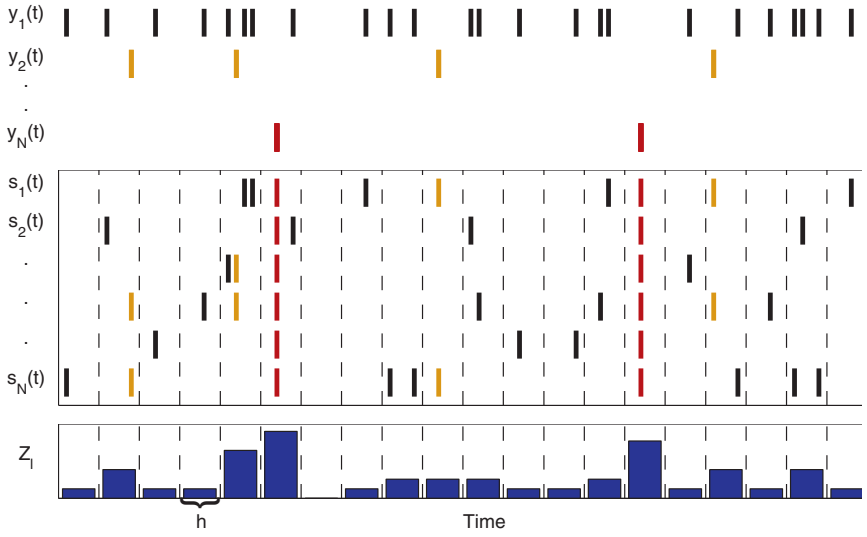

Figure 4.4: The mother process divided in the single components Y_i, used to compute its cumulants (modified from [37])

At the same time we can sample the variable Z from the parallel multi-dimensional recorded data, after that we have binned with a certain time resolution h. This means that we can construct some estimators for the cumulant $k_m(Z)$ and thanks to relation (4.3), construct hypothesis tests on the amplitude distribution and in particular the maximum order of correlation. The CuBIC method, as we will show in detail in the next section consists exactly in an iteration of such tests.

4.3.3 The test hypothesis

Now we have available all the elements to introduce the tests that are the core of the method.

The outcome of this analysis will be a lower bound $\hat{\xi}$ for the maximum order of correlation necessary to explain the data. The method consists in testing if a certain order of correlation m, measured from the data computing the m-th cumulant of the random variable Z from the binned data, is explained by a certain maximum order of correlation ξ.

We start describing in detail the method for $m = 2$ and then we generalize for a generic order of correlation.

Testing pairwise correlation

The analysis consists of a hierarchy of statistical hypothesis tests, each labeled by the variable ξ that correspond to the maximum order of correlation assumed under the null hypothesis. So, fixed the index ξ, the correspondent

null hypothesis $H_0^{2,\xi}$ states that the pairwise correlation ($m = 2$) in the data are compatible with the assumption that there is no correlation beyond the order ξ. Vice versa the alternative hypothesis $H_a^{2,\xi}$ states that an order of correlation larger then ξ is necessary to explain the measured pairwise correlation (e.g. $H_a^{2,4}$ implies that at least correlation of order 5 is necessary to generate in a MPP model the pairwise correlations measured from the data).

In order to formalize the statistics of the test it is important to clearly distinguish between the data on which perform the test and the model that provides the statistics against which the hypothesis is tested. In particular we introduce the variable Z' that describes the event count of the data on which the method is applied. In detail the object that we can sample from the binned multi-dimensional data set is a vector of realization $(z'_1, \ldots, z'_T)$. Each z'_i represents the count of event per bin of size h and $T = \frac{t_{stop} - t_{start}}{h}$. We use these samples to estimate the cumulants of Z'.

In parallel we take into account the general MPP model that we use to construct the statistics of the test. In particular we consider the random variable Z, consisting of the analogous event count of the model.

With this elements we can now formalize the null hypothesis $H_0^{2,\xi}$.

We consider the MPP model whose population event count Z has the maximal second cumulant, since we are testing using the pairwise correlation, under the assumption of first cumulant constrained by the estimation from the data samples and maximum order of correlation equal to ξ.

This requirements can be formalized stating that the MPP model underlying the null hypothesis is the solution of the following constrained maximization problem:

$$k^*_{2,\xi} := \max_{\lambda, A} \{K_2(Z)\}$$

Subjected to:

1. $K_1(Z) = K_1(Z')$
2. Maximum order of correlation of the multi-dimensional model equal to ξ

These two constrains in therm of parameter of the MPP(λ, A), that determines the population count Z, imply that $\lambda = \frac{k_1(Z')}{nh}$ and that $a_i = 0$ for $i = \xi + 1, \ldots, n$.

To solve the maximization problem we can now apply the result of theorem 4.3.2. It implies that:

$$k_2(Z) = \mu_2(\Xi)\lambda_M h = \sum_{l=1}^{n} l^2 a_l \lambda_M h$$

Using the equation already introduced in the previous section $\lambda_l = a_l \lambda_M$ and the second constrain of the problem ($a_i = 0$ for $i = \xi + 1, \ldots, n$), we obtain:

$$k_2(Z) = \sum_{l=1}^{n} l^2 \lambda_l h = \begin{pmatrix} 1^2 \\ \cdot \\ \cdot \\ \xi^2 \end{pmatrix} \cdot \begin{pmatrix} \lambda_1 \\ \cdot \\ \cdot \\ \lambda_\xi \end{pmatrix} h$$

where the dot denotes the standard scalar product. We can substitute this equation in the maximization problem using the vector notation $\bar{\xi_m} := (1^m, \ldots, \xi^m)^T$ and $\bar{\lambda_\xi} := (\lambda_1, \ldots, \lambda_\xi)^T$:

$$k^*_{2,\xi} := \max_{\bar{\lambda_\xi}} \{\bar{\xi_2} \cdot \bar{\lambda_\xi} h\}$$

subjected to $\bar{\xi_1} \cdot \bar{\lambda_\xi} h = k_1(Z')$.

This class of maximization problem, called Linear Programmed Problems, are uniquely solvable (e.g. using the Simplex Method). We can introduce the correspondent parameters of the model that solve the maximization problem: (λ^*, A^*).

We now use this measure to formalize the null hypothesis $H_0^{2,\xi}$. If we assume that the combination of rate and pairwise correlation measured in the data can be realized with maximum order of correlation $\leq \xi$, then the second cumulant of this population event count must be smaller than the general upper bound $k^*_{2,\xi}$. This allows to reformulate the null hypothesis of the test:

$$H_0^{2,\xi} : k_2(Z') \leq k^*_{2,\xi}$$

The alternative hypothesis

$$H_a^{2,\xi} : k_2(Z') > k^*_{2,\xi}$$

states that, assuming the framework of the MPP model, the pairwise correlations measured in the population imply the presence of correlation of order $> \xi$.

Test statistics and their distribution

The test introduced in the previous section requires to estimate the cumulant $k_1(Z')$ and $k_2(Z')$. From the data we can obtain the sample $(z'_1, \ldots, z'_T)$. We will estimate the cumulants by the standard sample mean and sample variance:

$$k_1 := \hat{k}_1(Z') = \frac{1}{T} \sum_{i=1}^{T} z'_i$$

$$k_2 := \hat{k}_2(Z') = \frac{1}{T} \sum_{i=1}^{T} (z'_i - k_1)^2$$

To test the null hypothesis $H_0^{2,\xi}$ we need the distribution of the test statistics k_2.

We first recall the general result that:

$$\mathbb{E}(k_2) = k_2(Z')$$

and:

$$\mathrm{Var}(k_2) \quad = \quad \frac{k_4(Z')}{T} - \frac{2k_2(Z')^2}{T-1}$$

Now to derive the distribution under the null hypothesis, we can use the fact that under$H_0^{2,\xi}$ Z' is the count events correspondent to the model that solves the maximization problem with parameter (λ^*, A^*). So we can compute all the cumulants of the variable Z' using the equation (4.3). We obtain the equation for the variance:

$$\mathrm{Var}(k_2) = (\frac{\mu_4(\Xi^*)\lambda_M^* h}{T} - \frac{(2\mu_2(\Xi^*)\lambda_M^* h)^2}{T-1})$$

If we assume a sample large enough, $T > 10000$, the distribution of k_2 under the null hypothesis is well approximated by a normal distribution. So we are now able to analytically compute a p-value for the null hypothesis $H_0^{2,\xi}$:

$$p_{2,\xi} = \int_{k_2}^{\infty} \frac{1}{\sqrt{2\pi \mathrm{Var}(k_2)}} \exp(-\frac{(t-k_{2,\xi}^*)^2}{2\mathrm{Var}(k_2)})dt$$

since we assume that the distribution of k_2 is normal of parameters $k_{2,\xi}^*$ and $\mathrm{Var}(k_2)$.

The rejection of the null hypothesis for a given ξ implies the assumption that there is no order of correlation beyond ξ. Thus the rejection of $H_0^{2,\xi}$ implies that $\xi + 1$ is a lower bound for the order of correlation. This suggest how to create a loop of tests iterated on ξ to infer the lower bound for the maximum order of correlation in the data. We will explicitly describe this procedure in the next section where we introduce the algorithm for the CuBIC method.

Testing higher order of correlation (m>2)

Now we extend the test to order different from two. It is possible to proceed exactly as in the case of $m = 2$ to obtain the formalization for the general null hypothesis $H_0^{m,\xi}$. The new maximization problem becomes:

$$k_{m,\xi}^* := \max_{\lambda, A}\{K_m(Z)\}$$

Subjected to:

1. $K_i(Z) = K_i(Z')$ for $i = 1, \ldots, m-1$
2. Maximum order of correlation of the multi-dimensional model equal to ξ

Using the vectorial notation we can re-write the maximization problem as:

$$k_{m,\xi}^* := \max_{\bar{\lambda_\xi}}\{\bar{\xi_m} \cdot \bar{\lambda_\xi} h\}$$

Differently from the case of $m = 2$ this problem is not always solvable. In particular the condition that make the solution unique is that $k_1(Z') < \cdots < k_{m-1}(Z')$.

In direct analogy to the case of pairwise correlation we can use the solution of this problem to reformulate the null hypothesis is function of an upper bound for the m-th cumulant:

$$H_0^{m,\xi} : k_m(Z') \leq k^*_{m,\xi}$$

Using the notions from the literature (Stuard and Ord 1987) on the cumulants it is possible to estimate the cumulants of the variable Z' up to m-th order and the variance $\mathrm{Var}(k_m)$. Under the hypothesis of normal distribution for $k_m$we can then compute the p-value:

$$p_{m,\xi} = \int_{k_m}^{\infty} \frac{1}{\sqrt{2\pi \mathrm{Var}(k_m)}} \exp(-\frac{(t - k^*_{m,\xi})^2}{2\mathrm{Var}(k_m)})dt$$

Implementation of the method

For the algorithm that implements the method constructing a nested loop iterated on the different order of correlation m and ξ we refer to the figure and to the original paper that introduced the method [42].

In the further application of the method in Chapter 5 we will use a version of the method fixing $m = 3$.

It has been developed also a non-stationary version of CuBIC. It has thought to deal in particular to deal with specific rate profile (e.g. cosine). Its construction is analogous to the stationary case, with the difference that all the analytics are derived taking into accountthe time-dependence of the rates. For the detail about this extension we refer to [41].

4.4 SPADE

The second method that we take into account in our comparative analysis is the Spike Pattern Detection and Evaluation (SPADE). This method, introduced in [43] in 2013, is a method that detect, with data-mining technique, and evaluate, with a statistical approach, the recurrence of specific patterns of synchrony. With pattern in this context we mean subset of margins indices. This method is constructed to detected whether there are, in a dataset of multiple parallel recorded neurons, units that synchronize their activity more than what expected by chance. This approach has been originally developed in the field of statistical neuroscience but it is a general tool to explore recurrent pattern in multi-dimensional point processes. This procedure does not return directly a result on the correlation order of the data, as CuBIC does, but more specific information on which margins are correlated, in the sense of synchronization of the events.

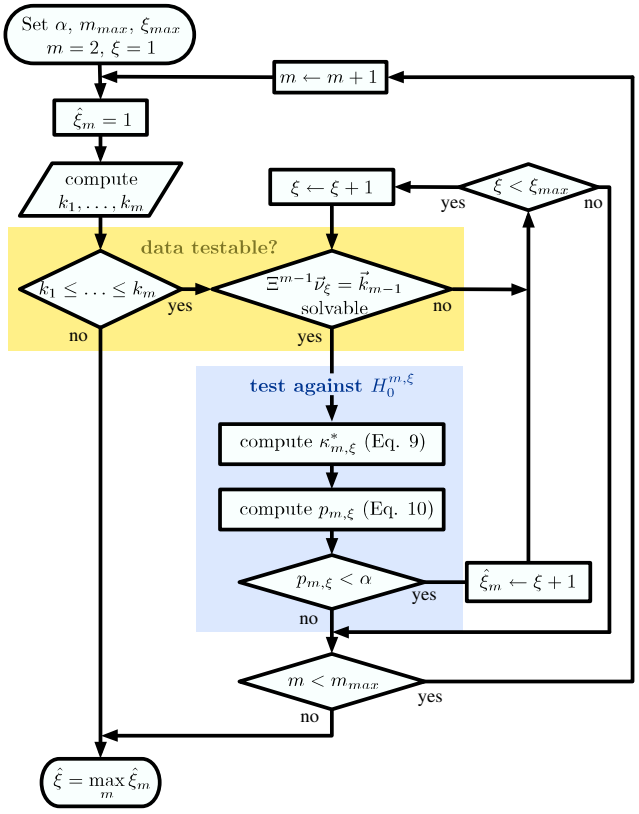

Figure 4.5: Algorithm to implement the recursive loop of test. We use two nested iterative cycles to test the different bound for the maximum order of correlation ξ, using as statistics the cumulants of the mother process up to order m (figure from [42])

The analysis of the interaction between the results of these different methods is the main focus of the last chapter of the thesis, in which we will use the model presented in Chapter 3 to test and compare these methods.

In the following sections we will briefly introduce the method, both the data-mining technique (FIM) and the statistical tests (PSF,PSR), for all the detail we refer to [43].

4.4.1 Frequent Itemset Mining

In order to describe the method we consider a discretized version of the data in the time interval $(t_{start}, t_{stop}]$. In our particular context we consider binned spike trains in the clipped version with a time resolution h (Section 4.1). Therefore we can consider the partition of time given by the bins $\Delta t_i = (t_{start} + (i-1)h, t_{start} + ih]$ for $i = 1, \ldots, T = \frac{t_{stop} - t_{start}}{h}$. Using the terminology of data-mining we call each margin, in our model each neuron, item. We define a transaction T_i as the set of items (neurons) for which an event occurred in the bin Δt_i. We call a pattern of synchrony a subset of one of the T_is with size bigger than 2. The support of a pattern is the number of transaction in which it occur, or equivalently the number of occurrences of a precise set of synchrony events.

A transaction of size $|T_i| = K$ has a possible number of subset (patterns) equal to $2^K - K - 1$. This means that the total number of patterns can largely exceed the number of transactions (i.e. time bins T).

In order to consider in the further analysis for the significance of occurrences only the non-trivial patterns, we select only the ones whose support (number of repetition) is larger then a minimum support. In our framework the minimal support is fixed equal to 2. All the pattern whose repetition exceed the minimum support are called frequent item set.

Furthermore all the patterns that have the same support of their superset are discarded. Indeed these are trivially explained by the repetition of the larger pattern containing the subset. The non-trivial frequent item sets are called closed frequent item set. The restriction of the analysis to the space of closed frequent item set $\mathfrak{C}$ does not imply any loss of information. Indeed the totality of frequent item set $\mathfrak{F}$ can be reconstruct from the set $\mathfrak{C}$:

$$\mathfrak{F} = \cup_{I \in \mathfrak{C}} \cup_{J \subset I, |J|>1} J$$

Given a set of transaction $(T_i)_{i=1}^T$ it is possible to extract the frequent item set with the Frequent Itemset Mining (FIM, [6]). Using this algorithm it is possible to extract the patterns of synchronous spikes, or in general margins of a multi-dimensional process, that occur frequently in the dataset (support ≥ 2) and are not trivially explainable because subset of other pattern (closed).

Applying this algorithm we end up with a set of list of indexes of neurons on which it is possible perform the statistical analysis to infer their significance under the hypothesis of independence. In the next sections we will present the

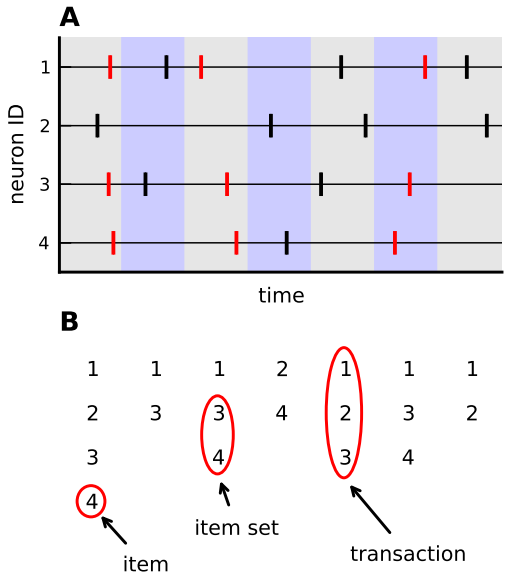

Figure 4.6: FIM: a binned multi-dimensional point point process and all its possible pattern of synchrony (modified from [43])

statistical tests constructed to detect which of these frequent item set occur more than what expected under the null hypothesis.

4.4.2 Pattern spectrum filtering

Now that we have available the set of all the closed frequent item patterns (CFIPs) we want to investigate their significance against the null hypothesis of independence. This hypothesis is equivalent to assume that the patterns occur multiple times just by chance.

Because of the big size of the set of samples (the CFIPs) we face a problem of multiple testing. For this reason, as proposed in [34], we pool the various patterns according their size z and their support, the number of repetitions, c in a two-dimensional histogram (pattern spectrum). Now instead of testing for the significance of each pattern we can test for the significance of a single signature formed by the couple (z, c).

In order to construct the statistical test we proceed using a Monte-Carlo approach using surrogate data (Section 4.2). In order to implement the null hypothesis H_0 of independence we repeatedly generate surrogate data of the original time series, in our context parallel spike trains. From each surrogate dataset we extract the CFIPs, using FIM, and we generate the correspondent pattern spectrum of all the possible signatures (z, c). The surrogates are gener-

ated in order to destroy the possible synchrony preserving other features such as rate of the events (e.g. spike shifting or spike dithering from Section 4.2).

In order to compute the p-values for each of all the possible signatures, we consider all the pattern spectrum of each set CFISs S_i of the i-th surrogate dataset. In particular we introduce the binary pattern spectrum P_i, defined for each couple (z, c):

$$P_i(z,c) := \begin{cases} 1 & \text{if } \exists A \in S_i : \text{sgt}(A) \succeq (z,c) \\ 0 & \text{otherwise} \end{cases}$$

In this formulation we use a the function sgt() that, given a pattern A, returns its signature (c_A, z_A). The $\succeq$ stands for the partial ordering on the real plane for which $(x', y') \succeq (x, y)$ if $x' \geq x$ and $y' \geq y$.

Averaging the binary spectra at each signature , we get the p-value spectrum $\hat{P}$:

$$\hat{P}(z,c) := \frac{\sum_{i=1}^{K} P_i(z,c)}{K}$$

in which K is the number of surrogates generated. $\hat{P}(z, c)$ yields an estimate of the probability to observe patterns with signature $(z', c') \succeq (z, c)$ under the null hypothesis H_0.

To asses the significance of any signature (z, c) we fix a significance level α. Since we want to perform m different hypothesis tests, one for each signature detected with FIM, we apply a Bonferroni correction. In each test we use a new significance level defined as $\alpha' := \frac{\alpha}{m}$ in order to overcome the problem of multiple testing and obtain a final level α for the significance of the entire method. Now any signature (z, c) for which the p-value $\hat{P}(z, c) > \alpha'$ is classified as significant. With this procedure (PSF) it is possible filter the patterns whose signature is significant.

The PSF tests the significance under the null hypothesis of independence, however it might fail to reject the patterns that result by chance from the combination of other actual patterns. In the next section we present the last part of the method that deals with this problem restricting the result to patterns that cannot be the sub-product of intersection or superposition of other assemblies.

4.4.3 Pattern set reduction

Let be $\mathfrak{P}$ the subset of CFISs reported as significant by PSF. We consider now a pair of patterns $(A, B) \in \mathfrak{P} \times \mathfrak{P}$ such that $B \subset A$ (therefore by definition of CFIS the occurences of the two patterns are so that $c_B > c_A$). The pattern set reduction consists of a series of different tests to check the significance of either A given B $(A|B)$ or B given A $(B|A)$. These tests can be applied to all the couples (A, B) of the class $\mathfrak{P}$, in order to get the refined class $\mathfrak{Q}$ of patterns which are mutually significant given each other. For the detail of these tests we refer to [43], in which the method has been developed.

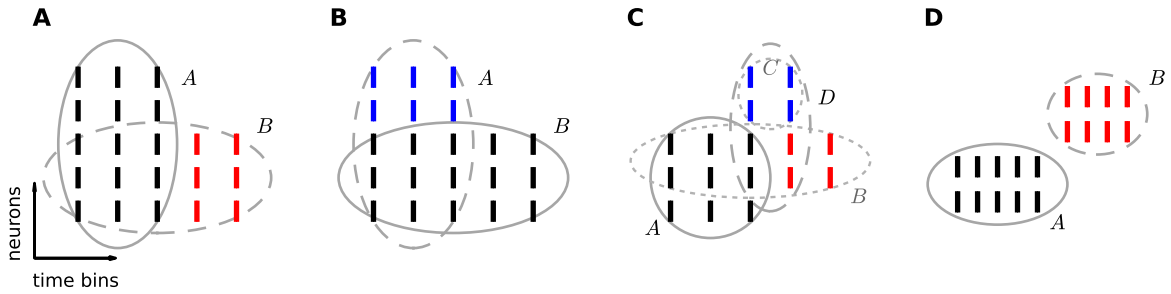

Figure 4.7: PSR: examples of false positive detected with PSF. The patterns grouped by the dashed ellipses are explainable given the true underlying patterns enlightened by different colors (modified from [43])

The entire procedure of SPADE (FIM+PSF+PSR) makes possible to detect patterns that occur more than what expected under the null hypothesis of independence and to infer their significance also taking into account all the rest of the data. Applying this method we end up with a list of indices of margins (or neurons) that we know to occur significantly.

In the next chapter we will try to integrate the results of the two methods here introduced using as test model the MPP described in Chapter 3. Since the two of them explore the correlation structure of the data with two different prospective we want to compare systematically their result and possibly to define an integrate workflow of data analysis. Since we have now available a wide selection of possible models with different feature we can also test the performances of the methods in parameters regime not explored in the original papers ([43],[42]).

Chapter 5

Comparative analysis of methods

In this chapter we present a comparative analysis of applications of the methods introduced in the previous chapter. We proceed by generating different datasets with different conditions of correlations and parameter settings. We apply systematically the methods to simulated data. The results of the methods are critically commented and analyzed in a comparative way. The data are simulated according to the model introduced in Chapter 3. In each simulation we emphasize different features of the models (e.g. stationarity and heterogeneity of rate, correlation structure). In cases where the inferred results are not interpretable in respect of the ground truth we investigate the possible causes that affected the tests (e.g. parameter scanning). The final purpose is to understand how to interpret the results by the different methods and then to make use of their properties for comparative analyses.. Because of simulated data we have control of all the statistics of the data. Hopefully this approach could lead to generate a work-flow that include the use of both methods in a comparative and iterative fashion.

First we analyze purely stationary data (Section 5.1) and then proceed to apply the methods to non-stationary data (Section 5.2). It has to be specified that the purpose of this work is not to get new results from data analysis but to test and, in more general, have a better comprehension of the two methods introduced. In particular we want to try to understand the relation between the results of the two methods in the most possible realistic scenario.

Both the technique for the simulations of the models and CuBIC method have been implemented during the in the framework of the thesis (in Python) [13].For the derivation of the results distribution for a large number of realizations it has been necessary to write specific scripts for the parallelization of the analysis.

5.1 Implementation of model and methods

In order to proceed with the comparative analysis we needed the implementation of both model and methods.

We implemented all the four generative algorithm for he MPP models in Python[13].

For the implementation of CuBIC method in Python we based on the Matlab [30] version provided by Staude et al.[42, 41]. In this implementation we fixed m, according to notations of the algorithm showed in Figure (4.5), equal to 3.

For the SPADE analysis we have used the existent Python implementation provided Torre et al. [43].

The implementations are also finalized to the ElectroPhysiology Analysis Toolkit (ElePhAnT) project[12]. The aim of this project is the construction of an open source platform to provide tools for the analysis of electrophysiological data.

In case of generation and analysis of large sample (e.g. $>$1000 realizations) we could access a High Performance Computing (HPC) cluster with 984 cores and 2.3 TB of memory. The parallelization of the analysis is a computational technique that consists of the simultaneous execution of a script on different cores of the same processor (e.g. HPC cluster). This technique allows to improve the computational performance and the duration of the elaboration. In particular for our case this means to have the possibility to generate and analyze in parallel all the 1000 realization of a specific model.

5.2 Stationary data

In this section we consider a test-cases in which the data are simulated according different models and different correlation structure. The generated data are unrealistic in relation to the real recorded data. The purpose here is not to simulate real data but to have a first impression of the performance of the methods with simulation in simplest parameter setting. We want, as far as possible, that the methods' results reflect the original correlation structure injected in the data. Here we consider simulation of stationary homogeneous model.

We will proceed presenting a table with several simulated datasets under different parameters conditions (Sections 5.1.1 and 5.1.2). Each of this datasets is analyzed with the methods in order to compare the results.

The results of the table are successively commented in detail. This first step has principally the purpose to get experienced in the interpretation of results that arise from very complex procedure as CuBIC and SPADE. In this first approach we have tried to present and interpret the data taking into account all the results obtained analyzing a few samples for each model..

We have realized that are very frequent unexpected and misleading results, in particular from CuBIC framework. So we decided to compute the distribution of CuBIC results for 1000 samples for each model in order to get a complete

information on the performance of the method and to be able to get a result comparable with the original findings of [42].

5.2.1 First set of data

All the data considered in Table 5.1 have the same parameters beside the correlation structure.

In particular the size of the dataset is 100 units. All the margins, according to the notation of multi-dimensional processes, are Poisson processes with stationary rate parameter $10Hz$. They are simulated in the time interval $(0s, 15s]$.

For both methods the width of the binning used to define the synchrony is $3ms$ and the significance level α is set to 0.05.

All the data considered are sampled from stochastic models. The result of the methods are sensible to this stochasticity. In a first approachwe consider the results obtained by a small number of realizations of the data.

We proceed here presenting the detail of all the data generated and the results obtained applying CuBIC and SPADE.

Data 1

The first dataset has been generated with all margins independent. As explained in Chapter 3 in our context this coincides with order of correlation equal to one.

Order of correlation equal to one is the result that we expect from the CuBIC analysis and that is obtained applying the method to the datasets. For some particular realizations the test could not be performed, because the data-set did not fit the analytical condition necessary to perform the test (ordered cumulant of the population count as explained in Chapter 4.3).

Also the SPADE analysis returns the result that we would expect. The independence of the data implies that no significant pattern is detected as correlated.

In this case the two results are coherent and this can be considered a useful double check for the correct implementation of the methods.

Data 2

The second dataset consists of a MPP model in which the correlations involve the entire population with order of correlation equal to 5. In particular this means that five margins of the multi-dimensional process randomly picked from the 100 that compose the multi-dimensional model are involved in each synchrony event. The amplitude distribution has been computed so that the total rate of synchrony is equal to $2Hz$. This implies that, on average, for the time period of 15 seconds, we expect on average 30 injections of synchrony involving each only 5 units.

Applying the CuBIC analysis we get the first unexpected result: there is high variability between different realizations and the tests infer, for many realizations, order of correlation equal to one, independent data. In general we know CuBIC to be a very conservative method because of the structure of the iteration cycle of test of hypothesis (see chapter 4). Since the result of independence in case of real data analysis is a case radically different from the simple

Data type	# units	Rate	Start-Stop	Binwidth	Synchronies	CuBIC res.	SPADE res.
1. Indep Poisson	100	10 Hz	0 s-15s	3 ms	No	1/failed	[]
2. MPP	100	10 Hz	0 s-15s	3 ms	Xi=5 Rate_coinc=2Hz	1/2/3/4	[]
2.* MPP	100	10 Hz	0 s-40s	3 ms	Xi=5 Rate_coinc=2Hz	3,4	[],Different patterns induced by the MPP
3. SIP	100	10 Hz	0 s-15s	3 ms	Xi=5 Rate_coinc=2Hz	1/2/3/4	[0,1,2,3,4] exact assembly injected
3*. SIP	100	10 Hz	0 s-40s	3 ms	Xi=5 Rate_coinc=2Hz	3,4	[0,1,2,3,4] exact assembly injected
4. MPP+indep Poiss	100	10 Hz	0 s-15s	3 ms	Xi=5 n_MPP=10 Rate_coinc=2Hz	1/2/3/4	Subsets of the assembly injected
5. MPP + SIP	100	10 Hz	0 s-15s	3 ms	xi_MPP=5 xi_sip=5 rate_sip=2Hz	4/5	Different patterns induced by the MPP but also the assembly injected in the SIP

Figure 5.1: Table of results 1

underestimation of the correlation, we try to improve this result before changing the length of the simulation (Data 2*) and in the next section computing the distribution of results for a large number of realization.

The result of SPADE are identical to the independent case, no pattern are detected as significant. But, while for CuBIC we are not able to immediately address the cause of the results, in this case the output is what we would expect. Indeed the margins which are involved in each of the synchrony are randomly picked among 100, each time independently. This implies that the probability to have same pattern of synchrony occurring more than one time is very small. Since we have fixed the minimal support equal to 2 (see chapter 4.4.1) we expect that single occurrence of pattern are not considered as frequent item set.

For many of the realizations, the two results are again coherent, not revealing evidence of correlation. But in this case they are misleading. We could expect such a result from SPADE with which we are looking for something that effectively is not present in the simulated data (frequent item sets). For what regards CuBIC the interpretation of the results is completely different. It is clear that it is not what we would expect and we have to explore the set of parameter to understand what mechanism has generate such result.

Data 2*

In order to try to answer to the questions generated in the analysis of data 2 we have generated the exact same model but for a longer simulation time. We generated the same model but for the interval of time $(0s, 40s]$.

We obtained that the order of correlation inferred by the CPP increases with the increasing of the length of the data. Increasing the time of simulation improve the results of the method at the same time without affectingthe result of SPADE. Indeed it keeps to return no significant pattern. These parameter dependency of CuBIC will be kept into account for further investigated in more general cases.

Data 3

In the third model the structure is similar to the previous dataset, with the difference that the margins that are involved into the synchrony are fixed. The same 5 units occur in each injection. In terms of the models introduced in chapter 3 this is realized by a multi-dimensional process with 95 independent Poisson processes and the remaining 5 units modeled with a SIP process. The SIP process has rate parameters ($\lambda = 10Hz, \lambda_c = 2Hz$). The total expected number of coincidences is the same of Data 2.

From the point of view of CuBIC analysis this formulation is equivalent to the dataset 2. Indeed, as explained in Chapter 4, this method take in consideration only the count of events per bin. The population histogram is supposed to have the same stochastic structure both in dataset 2 and 3, since the only difference is which margins are correlated, not the total amount of synchorny. This assumption is confirmed by the results that show the same variability across different realizations of the result for model 2, with many realizations inferred to be independent.

In the results of SPADE we expect a radical change from the previous case. Indeed this is exactly the the correlation structure that SPADE is constructed

to detect: the repetition of a particular sequence of margins (frequent item set). The method performs as expected, since the pattern inferred as occurring significantly is the one that compose the SIP model. No other pattern generated by chance alignment of the remaining independent units are detected.

Again it is difficult to interpret together the results of the two methods. While in the previous case we could be induced to suppose the data independent, in this case we can use SPADE results to detect the correlation structure. This would lead us to deepen the analysis of CuBIC exploring the parameter space, in particular in the case it is possible consider a longer time period of data or a smaller bin width.

Data 3*

With the same approach used before we changed the length of the simulation. We obtained again the result that CuBIC converge to the ground truth and the SPADE result, that was already satisfying, is invariant.

The difference is only in the fact that such approach can be suggested from the preliminary analysis also in real data analysis. Indeed even when the result of CuBIC is misleading we can use the information suggested by SPADE to point to further investigation. Instead in the previous condition (Data2) both the results, no pattern detected with SPADE and order of correlation equal to 1, mislead to assume the data to be uncorrelated.

Data 4

This dataset is close to the previous one but the correlation is spread around more units. It consists of 100 units in total, of which 90 are independent Poisson margins. The remaining 10 units are modeled as an MPP with amplitude correlation with maximum order of correlation equal to 5 and a rate of synchronous occurrences equal to $2Hz$. Each synchrony injection involves five unit but they are randomly picked among the 10 involved in the MPP. This implies that the patterns of synchrony of size 5 are randomly picked between the first ten units.

From the point of view of CuBIC once more the correlation structure it has not changed at all. The only difference with the previous datasets is which units are involved in synchrony and not the total amount of correlated events. The results confirm that again variability in the set $[1, 2, 3, 4]$. Again we face the problem oof many cases in which is inferred independence.

For SPADE we expect different results from both the previous cases. Indeed now the total amount of units involved in correlation is larger but the contribute of each is random. The output of the analysis once more reflects what we expect assuming the model. We get a bunch of different patterns all belonging to the correlated set of 10 units. Typically the significant patterns are subset smaller than the real injection of size 5. This model has been thought to simulate case in which a set of neuron present correlation but not all the units are involved into all the synchrony events (synaptic failure) or all the spikes are not correctly recorded.

We are again in a condition for which in many cases the interpretation of the results of CuBIC and SPADE together is difficult. Indeed in all the cases SPADE returns a certain correlation structure that leads to an incoherence in the cases in which CuBIC infers order of correlation equal to one.

Data 5

In the the last dataset the correlation structure is more complex. The data are generated by the superimposition of two different correlated models. The multi-dimensional set is composed by a MPP of dimension $n = 100$ with amplitude distribution $A = (a_1 = 0.91, a_2 = 0.01, \ldots, a_5 = 0.01, a_6 = 0, \ldots, a_{100} = 0)$ and heterogeneous rate parameter $\lambda = (\lambda_1 = 8Hz, \ldots, \lambda_5 = 8Hz, \lambda_6 = 10Hz, \ldots, \lambda_{100} = 10Hz)$. Afterwards the first five margins are superposed with a SIP model with rate parameters $(\lambda = 2Hz, \lambda_c = 2Hz)$. The final model has correlation order equal to five, randomly distributed across the 100 margins given by the MPP. The first five units have ulterior synchrony injections of size 5 with a rate of $2Hz$ (superposed SIP). The final marginal rates are $10Hz$ for all the components.

CuBIC is blind to distinguish between the different correlation structures, since it take in account only the total event count. The result is variable in a range of results close to the injected maximum order of correlation equal to 5. This result is different from all the previous ones and it is due to the higher frequency of synchrony induced by the amount of correlation both of the MPP and of the SIP. Even if we would expect such a result, it is not possible to distinguish the contribution to the inferred order of the two different models that have been superimposed.

The SPADE result is composed partly by patterns induced by the correlation of the MPP, each pattern is composed by random units different at each realization of the model, and the pattern of size 5 of the SIP model. In this case since we know the ground truth it is possible to distinguish these two class of patterns. For the application to real recorded data it would not be possible to distinguish patterns generated from such different mechanisms. At least not only with the SPADE result: ulterior investigations would be necessary. Our comparative work wants to go in the direction of using results from different methods to improve the capability of interpretation of the inferred correlation structures.

5.2.2 Second set of data

In order to get a better understanding of the result from the previous datasets we decided to to try to inject an higher order of correlation. In particular our purpose is to check whether the interpretation of results from CuBIC for the datasets 2 and 3 is biased by the choose of a low order of correlation (equal to 5).

All the data considered in table 5.2 have the same parameter of the previous one beside the correlation structure embedded in them, in particular we changed the order of correlation from 5 to 8. As the previous datasets the size is 100 units. Each of them is a marginally Poisson process with stationary rate parameter $10Hz$. They are simulated for in the interval $(0s, 15s]$. For both methods the width of the binning used to define the synchrony is $3ms$ and the significance level α is set to 0.05.

Data 1

Data type	# units	Rate	Start	Stop	Binwidth	Synchronies	CuBIC res.	SPADE res.
1. Indep Poisson	100	10 Hz	0 s	15 s	3 ms	No	1/failed	[]
2. MPP	100	10 Hz	0 s	15 s	3 ms	Xi=8 Rate_coinc=2Hz	5/6/7/8	Different patterns induced by the MPP
3. SIP	100	10 Hz	0 s	15 s	3 ms	Xi=8 Rate_coinc=2Hz	5/6/7/8	[0,1,2,3,4,5,6,7] exact assembly injected
4. MPP+indep poisson	100	10 Hz	0 s	15 s	3 ms	Xi=8 n_MPP=10 Rate_coinc=2Hz	5/6/7/8	Partial subsets of the assembly injected
5. MPP + SIP	100	10 Hz	0 s	15 s	3 ms	xi_MPP=10 xi_sip=5 rate_sip=2Hz	7/8	Different patterns induced by the MPP but also the assembly injected in the SIP

Figure 5.2: Table of result 2

In this data case nothing as changed in respect to Section 5.1.1 and all the results are analogous.

Data 2

The second dataset consists of a MPP model in which the correlations involve the entire population with order of correlation equal to 8, instead of 5 as before. In particular this means that in each synchrony event are involved 8 margins of the multi-dimensional process randomly picked from the 100 that compose the multi-dimensional model. The amplitude distribution has been computed such that the total rate of synchrony is equal to $2Hz$. This means that, on average, for the time period of 15 seconds we expect thirty injection of synchrony involving each only 8 units.

The results of CuBIC are variable, depending on the stochasticity of the data generated. The set of results is in the range that we would expect, given an underlying order of correlation equal to 8 and the conservative statistics of the method (chapter 4.3). Indeed we get results that varies in the set $(5, 6, 7, 8)$. This result has still a significant variability, but the most surprising aspect that we have revealed in the previous section (result equal to 1 also for correlated case) is not verified anymore. We will asses the problem of the variability in the next section but this is now a result that we could have expected from the structure of the method.

Applying SPADE we can infer as significant different patterns, but with a small number of repetition. This is again what we expect from such dataset. Indeed in each of the synchrony the units involved are randomly picked. This implies that the repetition of specific pattern is stochastic. The new result differs from the one of the previous section because a larger number of units are involved in the synchrony. This makes more likely the repetition of the same set of units.

These results, of CuBIC and SPADE, are coherent. In particular it is possible to access to different aspects of the correlation and to have a better understanding of the data. We can compare the size of the patterns inferred with the order of correlation resulting from CuBIC: they suggest exactly the presence of high order of correlation randomly distributed between the different units. This roughly describes the structure of the MPP, exactly the model underling the simulation.

Data 3

In the next dataset the structure is similar to the previous case, with the difference that the margins that are involved in the synchrony are fixed as like as in daset 3 in section 5.1.1. The units involved in the SIP model are now 8, instead of 5 in section 5.1.1.

The results od CuBIC show variability, due to the different stochastic configurations of each realizations, in the set of orders of correlation $(7, 8, 9)$. The interpretation of the change of the result in respect to the previous section it is completely analogous to the case of MPP (Data2).

As like as in section 5.1.1 the pattern detected by SPADE is composed by all and only the units involved in the SIP process, with the difference that now they are the first eight.

Once more the two results, of CuBIC and SPADE, agree. We can use this coherence to get more information about the data. Indeed the presence of repeated patterns revealed by SPADE is confirmed by the result of CuBIC and vice versa. This result leads to assume as underling model the SIP.

Data 4

Also for the fourth dataset we had reproduced the data as like in the first section, beside the difference of the order of correlation equal to 8. We have generated a MPP of size ten and amplitude distribution such that the maximum and unique order of correlation is eight. The amplitude distribution induces a rate of coincidences of $2Hz$ and the number of repetitions is on average 30. The remaining 90 units of the population are generated independently. This model implies that correlation of order 8 is spread randomly in a subset of 10 units.

The results of CuBIC, as expected, are analogous to the previous models. They vary between orders of correlation 6 and 8.

From the SPADE results we can reconstruct the entire assembly of correlated units. Indeed, since every injection involves 8 of them among the total of 10, it is very likely that all the units are assigned to one of the significant patterns.

These two results once more talk each other. Indeed from the high order of correlation inferred by CuBIC we can exactly expect a large number of pattern in the output of SPADE. From the inferred patterns we can also deduce the random nature of the correlation, spread among different units.

Data 5

Since the result of CuBIC were already stable for the case with order of correlation xi=5 we have not considered here the case with higher correlation order. It has still to be discussed the problematic of distinguish the correlation injected via the MPP and the one with the pure SIP. Because of the intrinsic nature of ambiguity between the two correlation it would be necessary to develop a specific framework to detect this difference. The hints that we obtain from this comparative analysis suggest that the integration of different methods for HOC can be a good approach for further development.

5.2.3 Detailed distribution of CuBIC results

Because of the difficulty occurred in the interpretation of the results, in particular of CuBIC, we decided to examine in depth the previous datasets in the same fixed parameter setting.

Another important reasons to better explore the results is the chance of using this analysis also to calibrate and to test the implementations of CuBIC method and of the generative algorithm for MPP models and to compare our results with the one obtained in [42].

In order to have a better knowledge about the performances of the method we decided to compute the distribution of the results for each of the 5 models described in the previous sections, assuming order of correlation equal to 5 and later on equal to 8. In this analysis we have generated 1000 different realizations of the 5 models and then applied CuBIC to each of them to get the distributions of the results.

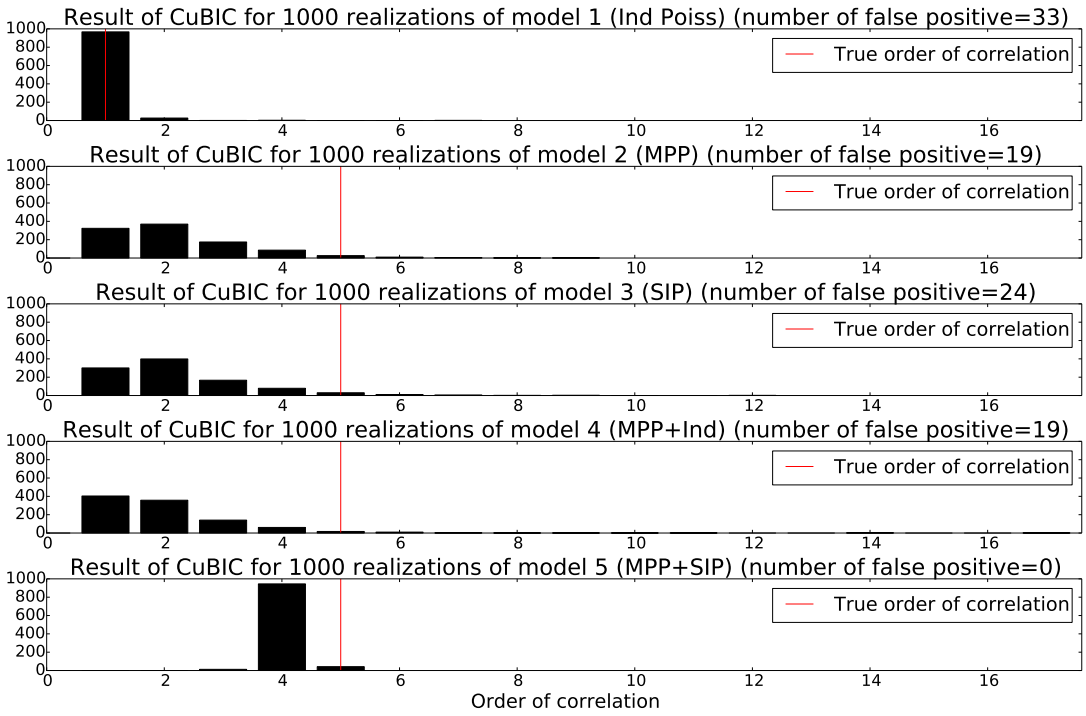

Figure 5.3: Distribution of the results inferred by CuBIC in 1000 different realization of the models described in section 5.1.1. The general parameter common to all the simulations are binsize=$3ms$, rate=$10Hz$, max order of correlation=5, time interval=$(0s, 15s]$

Order of correlation xi=5

The distributions showed in Figure 5.3 confirms the speculation that we did in section 5.1.1 using a small number of samples. For the model 1 (independence) the actual order of correlation is inferred in the most of the case; for model 2-3-4 we have approximately the same distribution of results and there is a considerable number of realization in which we inferred order equal to 1; for model 5 we have again a very good indication of the actual order of correlation 5, that we expect to be slightly (4 instead of the actual 5) underestimated because of the very conservative regime of the method.

Also the number of false negatives is aligned with what we expect. Indeed the significance level of the test is set to $\alpha = 0.05$. This implies that for a sample of size 1000 on average we should expect 50 false negative. For all the 5 models we got a lower number of false positive because of the very conservative approach used in CuBIC framework.

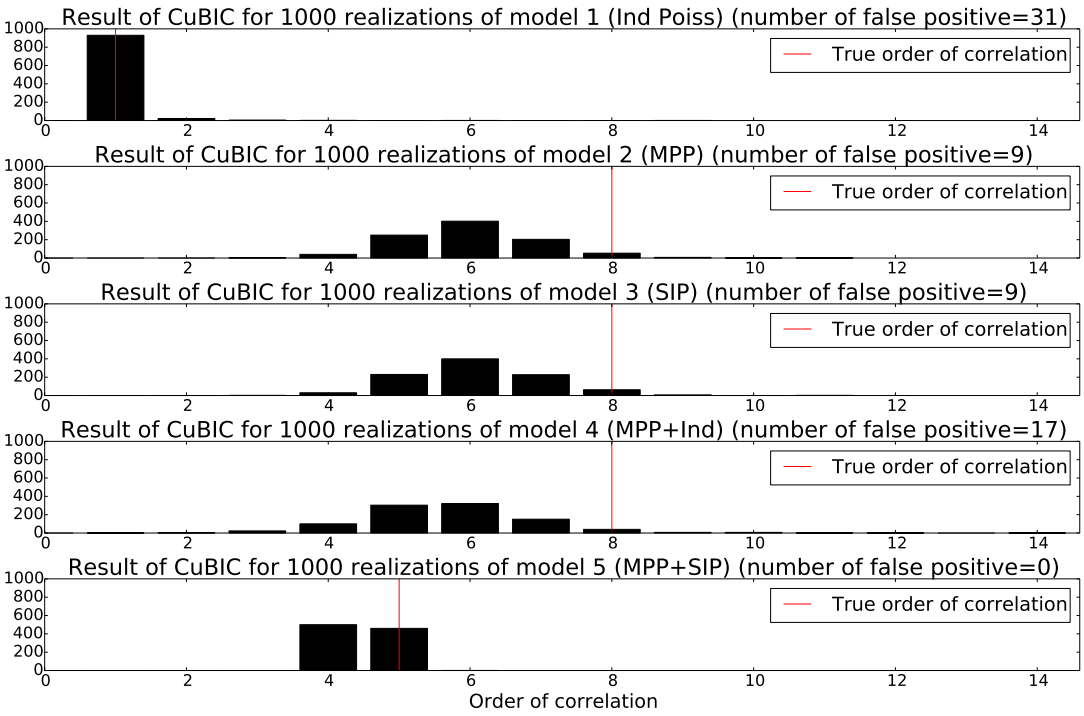

Figure 5.4: Distribution of the results inferred by CuBIC in 1000 different realizations of the models described in section 5.1.2. The general parameter common to all the simulations are binsize=$3ms$, rate=$10Hz$, max order of correlation=8, time interval=$(0s, 15s]$

In this context it is good to underline how is defined a false positive in the application of CuBIC. An order inferred by the method is considered a false positive if it is greater than the real maximum order of correlation of the data (in this case equal to 5). Indeed the null hypothesis tested in the framework of the method (using the notation of section in this application the null hypothesis is $H_0^{3,\xi}$) is that at least order of correlationξ or higher is necessary to explain the data. This implies that any order of correlation lower than the actual 5 is not considerable a false positive. It is exactly this nature of test that makes the method particularly conservative.

Order of correlation xi=8

We computed the same distribution of results for the case of correlation of order 8 (figure 5.4). The distributions confirm againthe first speculations presented in the previous section.

For the independent model nothing has changed.

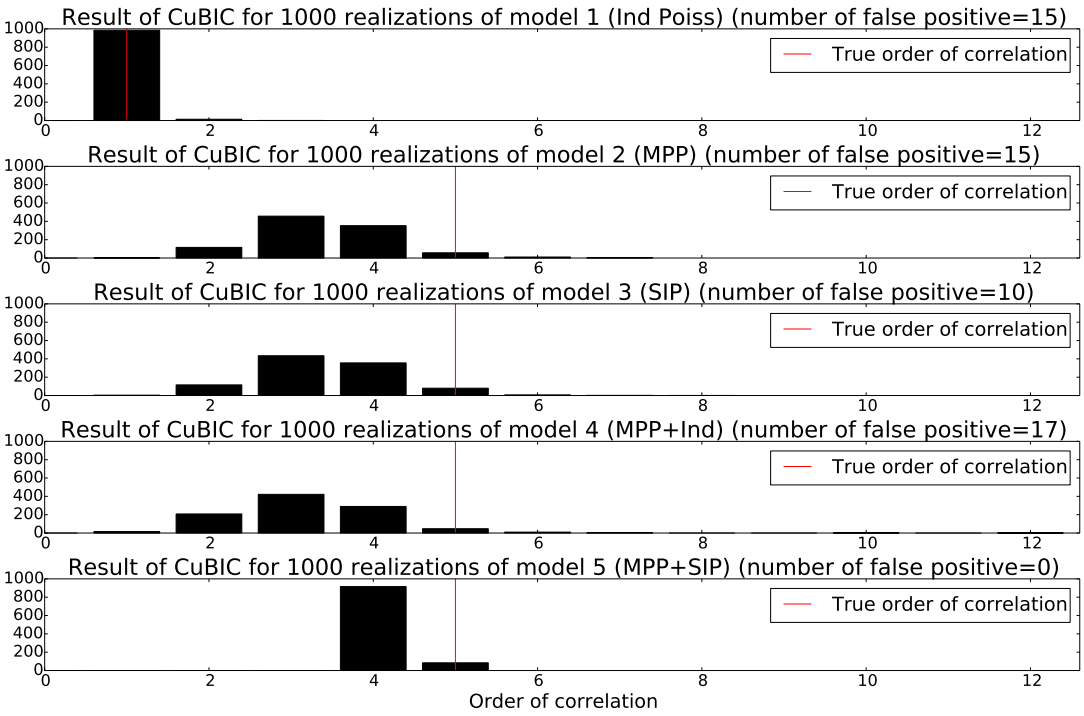

Figure 5.5: CuBIC result distribution. The general parameter common to all the simulations are binsize=$1ms$, rate=$10Hz$, max order of correlation=5, time interval=$(0s, 15s]$

For models 2,3,4 we get a distribution centered in 6. As we expect the actual order of correlation 8 is underestimated reflecting once more the conservative nature of the method.

The main difference between the distribution of results for model 2,3,4 is the skewness: in this case the distribution appears symmetric and it is very similar to the results presented in the original paper [42]. In order to improve the result also for correlation of order 5 we change the parameter of the bin width. In the next section we present this result with a bin size equal to $1ms$.

Model 5 is here constructed by the superimposition of an MPP with order of correlation equal to 5 and a SIP of size 8. The reflection on CuBIC result is the two pick amplitude showed in figure 5.4. In respect to the case where both the processes had order of correlation 5, we obtain an higher order of correlation for many realizations of the model.

Bin size correction for xi=5

In figure 5.5 we show the distribution of the results of CuBIC analysis for 1000 realizations. The models for the generation of the data are the same of the

previous sections with order of correlation equal to 5. The difference is the time precision used to bin the spike trains before the application of CuBIC. Now the bins-size used is of 1ms in respect to 3ms used before..

It is clear that the results have been improved. With this parameter setup in the case of xi=5 we get a symmetric distribution centered in 3. The main improvement is the low number of cases in which we get 1 as inferred order of correlation.

In the first section we decide to fix the bin width at $3ms$ because it is not realistic consider such a small time scale for the synchronization of the activity of the real neurons as explained in section 4.1.

5.3 Non-stationary data

The next set of data has been generated with the same purpose of the previous. The difference is the rate parameter. Now all the margins have non-stationary rate profiles instead of the single constant parameter λ. In particular the rate function is the same for all the datasets and for each margin. The rate is a step function defined in the interval of generation $(0s, 15s]$ as:

$$\lambda(t) = \begin{cases} 10Hz & \text{for } t \in [5s, 7.5s] \cup [12.5s, 15s] \\ 30Hz & \text{for } t \in (0s, 5s) \cup (7.5s, 12.5s] \end{cases} \tag{5.1}$$

All the other parameters are fixed as in the previous cases. The data are binned with a time precision of $3ms$. The total number of units is 100. The five different datasets present five different correlation structures, each analogous to the stationary case.

We proceed generating the same five models presented in the previous section, with the only different of non-stationary rate profile $\lambda(t)$. The order of correlation is fixed at 8 and each model has a different correlation structure explained in Section 5.2.2.

We do not present in detail the results of each different model since they are surprisingly similar.

Indeed CuBIC analysis returns in most of the realizations order of correlation 4 for all five models. This is unexpected in the case of model 1 in which each component was generated independently: CuBIC returns always false positives for this specific model.

Also in the case of correlated data, where we expect an order of correlation close to the actual maximum order of correlation (xi=8), the inferred order is 4.

The explanation is that the method is biased by the rate modulation. In the case of independent data the time locking of the rate variation determines an order of correlation not effectively injected in the simulation. At the opposite for correlated data (model 2-3-4-5) the rate modulation affects the result hiding the actual injection of synchrony.

Data type	# units	Rate	Start	Stop	Binwidth	Synchronies	CuBIC res.	Cubic non-stat (gamma)	SPADE res.
1. Indep Poisson	100	10 Hz in [0,5s] and [7.5s,12.5s] 30Hz in [5s,7.5s] and [12.5s,15s]	0 s	15 s	3 ms	No	4	1	[]
2. MPP	100	10 Hz in [0,5s] and [7.5s,12.5s] 30Hz in [5s,7.5s] and [12.5s,15s]	0 s	15 s	3 ms	Xi=8 Rate_coinc=5 Hz	4	1	[]
3. SIP	100	10 Hz in [0,5s] and [7.5s,12.5s] 30Hz in [5s,7.5s] and [12.5s,15s]	0 s	15 s	3 ms	Xi=8 Rate_coinc=5 Hz	4	1	[0,1,2,3,4,5,6,7] exact assembly injected
4. MPP+Indep Poiss	100	10 Hz in [0,5s] and [7.5s,12.5s] 30Hz in [5s,7.5s] and [12.5s,15s]	0 s	15 s	3 ms	Xi=8 n_MPP=10 Rate_coinc=5 Hz pi=0.8	4	1	Incomplete partial pattern
5. CPP + SIP	100	10 Hz in [0,5s] and [7.5s,12.5s] 30Hz in [5s,7.5s] and [12.5s,15s]	0 s	15 s	3 ms	xi_cpp=8 xi_sip=5 rate_sip=5Hz	4/5	1	Different patterns induced by the CPP but also the assembly injected in the SIP

Figure 5.6: Table of result 3

The SPADE analysis seems to deal with the non-stationarity. Indeed the use of surrogates, as explained in Section 4.1, keeps into account the local variation of rates. In particular it is possible to get the same results as in the stationary regime: no patterns detected for model 1-2, the pattern forming the SIP process in model 3, partial patterns of the MPP in 4 and for model 5 many patterns of the underlying MPP beside the five units involved in the superimposed SIP. The interpretation of the results are equivalent to the stationary case treated in 5.1.2.

The biased results of CuBIC prevent to proceed with any comparative analysis. In the next section, as already done for the stationary case, we proceed analyzing in detail the distribution of result of CuBIC for a larger sample of realization.

5.3.1 Detailed distribution of CuBIC results for non-stationary data

In this section, as like as in 5.1.3 for stationary data, we present the distribution of results of CuBIC for 1000 different realizations of the 5 models . The actual correlation order for models 2-3-4-5 is equal to 8.

From the distribution, showed in 5.7, we get the expected result given what presented in the previous section for a small number of samples. Indeed for all the models the majority of realizations shows order of correlation 4, also in the case of independent data. We deduce that this results are affected from the time variability of the rates.

For all the 1000 realizations of model 1, in which the data are generated independently, CuBIC return a false positive. Indeed all the order of correlation inferred is higher than 1.

As already explained the results for models 2-3-4-5 are neither false positive or false negative. Indeed the actual order of correlation is higher than the lower bound inferred by the methods in all the cases. However the methods performance can be considered less accurate than the stationary case in which the order of correlation was closer to the actual value (8). Since we know that also in the case of independent data the result is 4 we can not be sure if the output of CuBIC for models 2-3-4 is due to the actual correlation or only to the non-stationary rates.

Binsize correction

As like as for the stationary case we try to improve the results using a finer discretization for the binning. We fix the bin size at $1ms$. The results that we obtain confirm that the output is driven by the non-stationary rate more than the actual correlation. Indeed for the independent case we obtain a lower order of correlation, now order 3 is inferred instead of 4, but also for all the correlated sets of data (models 2-3-4-5) we get a lower bound for the maximum order of correlation.

Our interpretation is now that the smaller time scale used let be the method less sensible to the modulation of rate, but at same time this effect does not enlighten the actual correlation injected in the simulations.

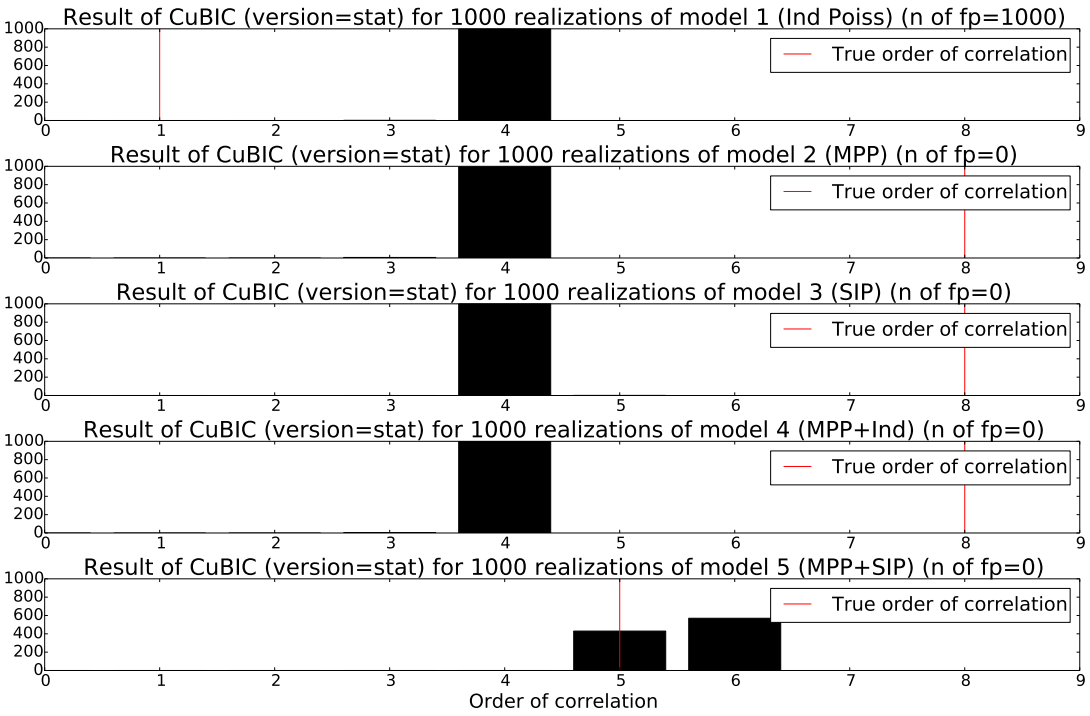

Figure 5.7: CuBIC results distribution. The general parameter common to all the simulations are binsize=$3ms$, rate=$\lambda(t)$ from (5.1), max order of correlation=5, time interval=$(0s, 15s]$

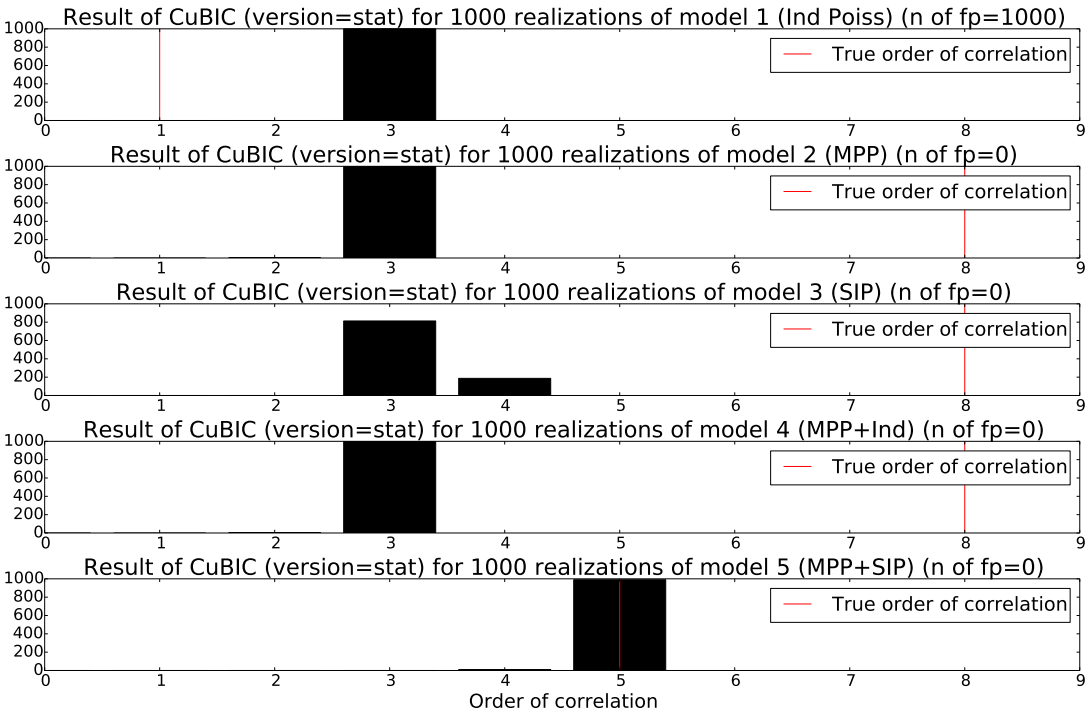

Figure 5.8: CuBIC results distribution. The general parameter common to all the simulations are binsize=$1ms$, rate=$\lambda(t)$ from (5.1), max order of correlation=5, time interval=$(0s, 15s]$

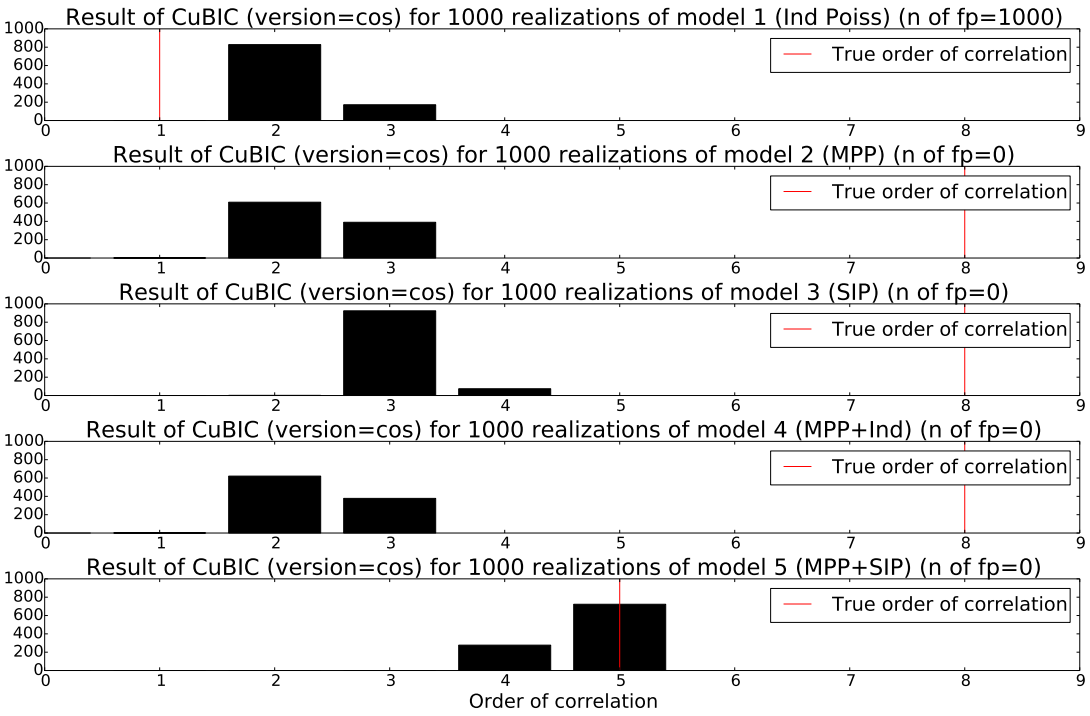

Figure 5.9: Cosine version of CuBIC: results distribution. The general parameter common to all the simulations are binsize=$1ms$, rate=$\lambda(t)$ from (5.1), max order of correlation=5, time interval=$(0s, 15s]$

Non-stationary version of CuBIC

Because of the large number of false positive encountered in this case of non-stationary rate we decided to implement the non-stationary version of CuBIC presented in [41]. In particular, according to the terminology introduced in the paper, we implemented three different versions of the method to deal respectively with cosine, uniform and gamma distributions of rates.

In equation (5.1) we intentionally choose a distribution of rate not considered in [41] because the original purpose here was not to test the performance of the method and reproduce the results of the paper. Instead we want to test the method in a more general context and in particular to integrate the interpretation of the result with SPADE. So we have decided to use the original version that does not assume non-stationary rate.

Because of the difficulties in the interpretation of the results presented in the previous section we decided in a second moment to check whether it is possible to obtain a better result with the extension of CuBIC proposed in [41].

The results obtained with the three different version of non-stationary rate

distribution do not lead to the expected results. As shown in Figure (5.9) for the case of the assumption of the cosine distribution for the rates, the methods return again a large number of false positive for the model 1 and even lower correlation order for the remaining models. The assumption in the method of a rate distribution different from the one used to generate the data biased the results of CuBIC analysis.

Now it is clear that in the case of application of CuBIC to non-stationary data it is necessary the knowledge of the exact distribution of the rate profile of the data. The problem of estimation of underlying rate from the real data is not trivial and is matter of study in the field ([32]).

The purpose of the thesis is not to improve or to extend the existent methods but to evaluate their performance in different contexts and for different models. The intent is not to reproduce the results presented in the original papers, but to test the method out of the conditions for which it was originally built.

Also providing an adaptation of the method to be applied to the specific set of data considered in this simulation, it would not provide a solution for the general and high variable rate profiles that can occur in real data.

Our purpose is to enlighten the difference between an analytical framework as like as CuBIC that is constrained to accessibility of the macroscopic statistics of the data (e.g. rate distribution) and a method that constructs the statistics with Monte Carlo methods as SPADE, that is intrinsically built to take into account the local variability of the data.

Summary and Outlook

We presented a generalized parametric model (MPP model) to generate correlated multi-dimensional Poisson Processes (Chapter 3). This is a generic multi-dimensional point process but here it has been thought to model and simulate electrophysiological correlated time series (spike trains) and the correlation is generated via the injection of synchronous events. In this context each marginal process models one neuron. We derived specific algorithm for the simulation of the model with different parameter conditions. In particular we elaborated the analytical descriptions and the generative algorithm for four different firing rates settings varying both in time and marginal processes:

1. stationary homogeneous across marginal processes rate parameter λ
2. stationary heterogeneous rates $(\lambda_1, \ldots, \lambda_n)$
3. non-stationary in time homogeneous rates $\lambda(t)$
4. non-stationary and heterogeneous rate $(\lambda_1(t), \ldots, \lambda_n(t))$

With this model it is possible to simulate data with different correlation structures. It is possible to induce different orders of correlations, injecting synchrony of different sizes. The neurons involved in the synchrony can be randomly selected between all the marginal processes or involve only specific sub-sets of neurons. A particular case of this framework is the single interaction model (SIP), already used to simulate correlated spike trains (e.g. in [27]), in which in all the synchrony are involved all the marginal processes.

We proceed in Chapter 4 introducing two different analyses of higher-order correlations for massively parallel spike trains: CuBIC ([42]) and SPADE ([43]).

CuBIC is a statistical method that consists in series of statistical tests to infer a lower bound for the maximum order of correlation in parallel spike trains. The statistics of the test are computed taking into account only population measure, as like the population histogram (Section 4.1).

On the other hand SPADE detects repetition of specific patterns of synchrony neurons and evaluate via Monte-carlo techniques the significance of occurrences in respect to the null hypothesis of independence. The investigation performed by SPADE is at level of single component and the results consist in specific set of neurons that are synchronized significantly more than by chance.

The two methods investigate different aspects of the correlation structure. The relation between the results of each of them is not immediate and it is not know how they perform when different correlation structures are embedded in the same data. In chapter 5 we introduce a comparative analysis of the two methods to investigate these questions.

We used the MPP model to simulate parallel spike trains with different marginal properties (that are, different firing rate profiles and ISI distributions) and correlation structures. We separately applied both CuBIC and SPADE to each data set, and compared the results. The comparison allowed us in most cases to better reveal the real correlation structure underlying the simulated data as compared to when only one of the two methods was applied. In some cases the combination of results is not possible, because the complexity of the correlation structure and the parameter setting bias one of the two method or make impossible to relate the information obtained by the two methods. In such cases we performed deeper investigations about the performances of the single method, eventually exploring the parameter settings to improve the results. In particular we have computed the distribution of the results of CuBIC from 1000 different realizations of the same model with different parameters settings in order to obtain a precise measure of the performance of the method for our simulated data.

We plan to extend the current study in two directions. On the one hand we aim at including further methods for HOC detection in the comparison. Interesting methods in this respect are presented in [5, 15]. The first of them uses different statistical measures (e.g. pairwise correlations) in order to detected sub-population of neurons highly inter-correlated. The second one is a visual method thought to detect in the data the particular spetio-temporal structure of synfire-chains. The extension of the comparative analysis is interesting for both methods since they investigate for different aspects of the correlation structure the same class of data of CuBIC and SPADE.

On the other hand we plan to investigate a wider variety of data sets, both with respect to their correlation structure and to their marginal properties. Regarding the latter point, we want to extend the MPP model to more realistic non-Poisson spike trains.

The MPP can be also used to generate correlated -rather than independent, as classically done - surrogates of the original data, retaining some of the original data's marginal properties (e.g. their firing rate profiles) while enforcing a desired correlation structure. Such surrogate data sets can then be used via a Monte-Carlo approach to test whether the data match the specified null hypothesis. This idea will be part of future work.

Acknowledgements

I would like to express my gratitude to prof. Cristina Zucca and to prof. Sonja Grün that gave me the opportunity to develop my thesis at INM6 in the Jülich Forschungszentrum. Their valuable supervision was indispensable to accomplish the work.

My gratitude goes also and not less to Emiliano Torre for his constant and indefatigable help, without him this thesis would have not be possible.

I would also like to thanks prof. Markus Diesmann and prof. Sonja Grün and all my fellow researcher at INM6 for have welcomed me since the first day. The suggestions and the experience that they shared with me has been priceless.

Bibliography

[1] Moshe Abeles. Role of the cortical neuron: integrator or coincidence detector? *Israel journal of medical sciences*, 18(1):83–92, 1982.

[2] Moshe Abeles. *Corticonics: Neural circuits of the cerebral cortex*. Cambridge University Press, 1991.

[3] Moshe Abeles and George L Gerstein. Detecting spatiotemporal firing patterns among simultaneously recorded single neurons. *Journal of Neurophysiology*, 60(3):909–924, 1988.

[4] Denise Berger, Christian Borgelt, Sebastien Louis, Abigail Morrison, and Sonja Grün. Efficient identification of assembly neurons within massively parallel spike trains. *Computational intelligence and neuroscience*, 2010:1, 2010.

[5] Denise Berger, David Warren, Richard Normann, Amos Arieli, and Sonja Grün. Spatially organized spike correlation in cat visual cortex. *Neurocomputing*, 70(10):2112–2116, 2007.

[6] Christian Borgelt. Frequent item set mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(6):437–456, 2012.

[7] Emery N Brown, Riccardo Barbieri, Valérie Ventura, R Kass, and L Frank. The time-rescaling theorem and its application to neural spike train data analysis. *Neural computation*, 14(2):325–346, 2002.

[8] Emery N Brown, Robert E Kass, and Partha P Mitra. Multiple neural spike train data analysis: state-of-the-art and future challenges. *Nature neuroscience*, 7(5):456–461, 2004.

[9] György Buzsáki. Large-scale recording of neuronal ensembles. *Nature neuroscience*, 7(5):446–451, 2004.

[10] David Roxbee Cox and Valerie Isham. *Point processes*, volume 12. CRC Press, 1980.

[11] DJ Daley and D Vere-Jones. An introduction to the theory of point processes, volume i: Elementary theory and methods of probability and its applications, 2003.

[12] M. Denker, A. Yegenoglu, D. Holstein, E. Torre, T. Jennings, A. Davison, and S. Grün. elephant: An open-source tool for the analysis of electrophysiological data. *Proceedings of the 11th Meeting of the German Neuroscience Society, Neuroforum 2015: T27-2B*, 2015. Project website: https://github.com/NeuralEnsemble/elephant.

[13] F.L. Drake (eds) G. van Rossum. *Python Reference Manual*. PythonLabs, 2001. Available at http://www.python.org.

[14] Apostolos P Georgopoulos, John F Kalaska, Roberto Caminiti, and Joe T Massey. On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex. *The Journal of Neuroscience*, 2(11):1527–1537, 1982.

[15] George L Gerstein, Elizabeth R Williams, Markus Diesmann, Sonja Grün, and Chris Trengove. Detecting synfire chains in parallel spike data. *Journal of neuroscience methods*, 206(1):54–64, 2012.

[16] Wulfram Gerstner and Werner M Kistler. *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press, 2002.

[17] EM Glaser and WB Marks. On-line separation of interleaved neuronal pulse sequences. *Data Acquisition Process Biol Med*, 5:137–156, 1968.

[18] Sonja Grün, Markus Diesmann, and AMHJ Aertsen. Unitary events in multiple single-neuron spiking activity: I. detection and significance. *Neural Computation*, 14(1):43–80, 2002.

[19] Sonja Grün and Stefan Rotter. *Analysis of parallel spike trains*. Springer Series in Computational Neuroscience, 2010.

[20] D Hebb. The organization of behaviour: A neuropsychological theory. 1949.

[21] David H Hubel and Torsten N Wiesel. Receptive fields of single neurones in the cat's striate cortex. *The Journal of physiology*, 148(3):574–591, 1959.

[22] David H Hubel and Torsten N Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106–154, 1962.

[23] DH Hubel and TN Wiesel. Shape and arrangement of columns in cat's striate cortex. *The Journal of physiology*, 165(3):559–568, 1963.

[24] Eric R Kandel, James H Schwartz, Thomas M Jessell, et al. *Principles of neural science*, volume 4. McGraw-Hill New York, 2000.

[25] Robert E Kass, Uri T Eden, and Emery N Brown. *Analysis of neural data*. Springer, 2014.

[26] Peter König, Andreas K Engel, and Wolf Singer. Integrator or coincidence detector? the role of the cortical neuron revisited. *Trends in neurosciences*, 19(4):130–137, 1996.

[27] Alexandre Kuhn, Ad Aertsen, and Stefan Rotter. Higher-order statistics of input ensembles and the response of simple model neurons. *Neural Computation*, 15(1):67–101, 2003.

[28] Sebastien Louis, George L Gerstein, Sonja Grün, and Markus Diesmann. Surrogate spike train generation through dithering in operational time. *Frontiers in computational neuroscience*, 4, 2010.

[29] Eugene Lukacs. *Characteristic functions*, volume 5. Griffin London, 1970.

[30] MATLAB. *version 7.10.0 (R2010a)*. The MathWorks Inc., Natick, Massachusetts, 2010.

[31] Edwin M Maynard, Craig T Nordhausen, and Richard A Normann. The utah intracortical electrode array: a recording structure for potential brain-computer interfaces. *Electroencephalography and clinical neurophysiology*, 102(3):228–239, 1997.

[32] Martin Nawrot, Ad Aertsen, and Stefan Rotter. Single-trial estimation of neuronal firing rates: from single-neuron spike trains to population activity. *Journal of neuroscience methods*, 94(1):81–92, 1999.

[33] Donald H Perkel and Theodore H Bullock. Neural coding. *Neurosciences Research Program Bulletin*, 1968.

[34] David Picado-Muiño, Christian Borgelt, Denise Berger, George Gerstein, and Sonja Grün. Finding neural assemblies with frequent item set mining. *Frontiers in neuroinformatics*, 7, 2013.

[35] Sidney C Port. *Theoretical probability for applications*, volume 206. Wiley-Interscience, 1994.

[36] Murray H Protter, B Charles Jr, et al. *A first course in real analysis*. Springer Science & Business Media, 1991.

[37] Imke CG Reimer, Benjamin Staude, Werner Ehm, and Stefan Rotter. Modeling and analyzing higher-order correlations in non-poissonian spike trains. *Journal of neuroscience methods*, 208(1):18–33, 2012.

[38] Edward M Schmidt. Computer separation of multi-unit neuroelectric data: a review. *Journal of neuroscience methods*, 12(2):95–111, 1984.

[39] Edward M Schmidt. Instruments for sorting neuroelectric data: a review. *Journal of neuroscience methods*, 12(1):1–24, 1984.

[40] Wolf Singer. Neuronal synchrony: a versatile code for the definition of relations? *Neuron*, 24(1):49–65, 1999.

[41] Benjamin Staude and Stefan Rotter. Higher-order correlations in non-stationary parallel spike trains: statistical modeling and inference. *BMC Neuroscience*, 10(Suppl 1):P108, 2009.

[42] Benjamin Staude, Stefan Rotter, and Sonja Grün. Cubic: cumulant based inference of higher-order correlations in massively parallel spike trains. *Journal of Computational Neuroscience*, 29(1-2):327–350, 2010.

[43] Emiliano Torre, David Picado-Muiño, Michael Denker, Christian Borgelt, and Sonja Grün. Statistical evaluation of synchronous spike patterns extracted by frequent item set mining. *Frontiers in computational neuroscience*, 7, 2013.

[44] James Trousdale, Yu Hu, Eric Shea-Brown, and Krešimir Josić. A generative spike train model with time-structured higher order correlations. *Frontiers in computational neuroscience*, 7, 2013.

[45] Henry C Tuckwell. Introduction to theoretical neurobiology, linear cable theory and dendritic structure. *Cambridge studies in mathematical biology.*, 1988.